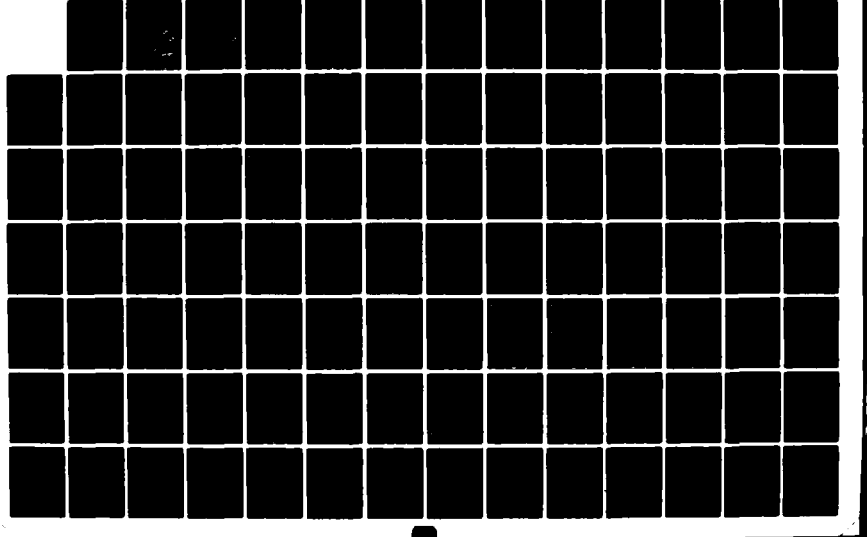
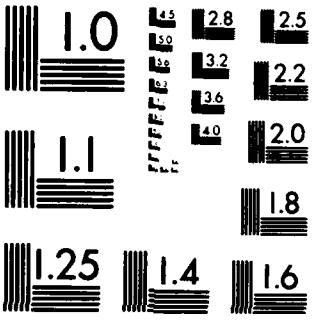
AD-A127 695 THE COMPLEAT TRAIDSMAN(U) GENERAL RESEARCH CORP SANTA 1/3
BARBARA CA T PLAMBECK 01 SEP 69 GRC-IM-711/2

UNCLASSIFIED F/G 9/2 NL





IM-711/2

INTERNAL MEMORANDUM

THE COMPLEAT TRAIDSMAN

by

Tom Plambeck

September 1968

(Revised September 1969)

The subject document is complete. There are no missing pages.

GENERAL RESEARCH CORPORATION
P. O. BOX 3587, SANTA BARBARA, CALIFORNIA 93105

Research reported in this document was originated through independent efforts, not under a Government contract or program.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: IM-711/2	2. GOVT ACCESSION NO.: AD-A127695
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): THE COMPLEAT TRAIDSMAN	5. TYPE OF REPORT & PERIOD COVERED: FINAL, 1 Sep 68 - 1 Sep 69
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): TOM PLAMBECK	8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: General Research Corporation, P.O. Box 3587, Santa Barbara CA 93105	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 305810UU
11. CONTROLLING OFFICE NAME AND ADDRESS: Rocket Propulsion Laboratory/DY, Stop 24, Edwards AFB CA 93523	12. REPORT DATE: 1 Sep 69
	13. NUMBER OF PAGES: 304
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited distribution; for more information, contact AFRPL/TSPR, Stop 24, Edwards AFB CA 93523.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Traid, trajectory, subroutines, matrix, vector.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Traid is a family of subroutines which handle the calculation of powered and guided trajectories and of Keplerian orbits. There are subroutines which integrate vehicles forward or backward in time, with thrust-mass-lift-drag-response input by the user, with either preassigned or computed guidance commands. Orbits may be elliptical (multiple revolutions are allowed) or hyperbolic; transfer between integrated flight and orbital representation is accomplished automatically. (Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

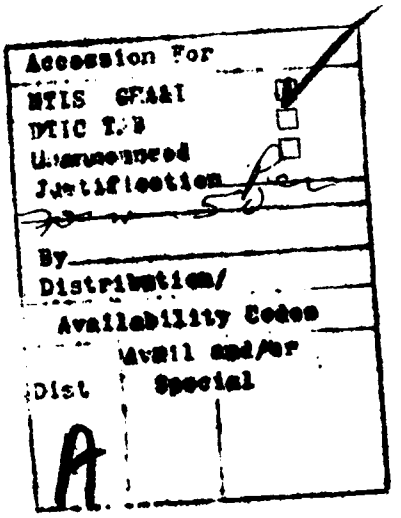
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (CONTINUED)

In addition to trajectory calculations, TRAID also provides assistance in the following areas: (1) card-data input, (2) printed output, with standardized formats, (3) vector- and matrix-manipulating routines, (4) problem supervision -- e.g., printing title page, enforcing time and page limits, and (5) miscellaneous aids -- e.g., plotting, manipulating tabular data.

If you have a trajectory problem, TRAID can make your programming task easier in two ways. First, your program will accomplish more per FORTRAN statement. The second (hidden) benefit is that your program will probably contain fewer errors, because your attention can remain concentrated on the problem as distinct from the spelling.

To use TRAID, you must write a main program in FORTRAN. In this program you must worry about two things: (1) the allocation of storage for arrays, and (2) calling the appropriate TRAID routines to perform the desired operations.



This document does not have an AFRPL-TR number per Ms. Chalfant, AFRPL/STINFO

CONTENTS

1.0 TRAID -- WHAT AND WHY?

TRAID is intended to help solve trajectory problems.

TRAID is a family of subroutines which handle the calculation of powered and guided trajectories and of Keplerian orbits. There are subroutines which integrate vehicles forward or backward in time, with thrust-mass-lift-drag-response input by the user, with either preassigned or computed guidance commands. Orbits may be elliptical (multiple revolutions are allowed) or hyperbolic; transfer between integrated flight and orbital representation is accomplished automatically.

In addition to trajectory calculations, TRAID also provides assistance in the following areas:

- card-data input
- printed output, with standardized formats
- vector- and matrix-manipulating routines
- problem supervision -- e.g., printing title page, enforcing time and page limits
- miscellaneous aids -- e.g., plotting, manipulating tabular data.

If you have a trajectory problem, TRAID can make your programming task easier in two ways. First, your program will accomplish more per FORTRAN statement. The second (hidden) benefit is that your program will probably contain fewer errors, because your attention can remain concentrated on the problem as distinct from the "spelling."

To use TRAID, you must write a main program in FORTRAN. In this program you must worry about two things: (1) the allocation of storage for arrays, and (2) calling the appropriate TRAID routines to perform the desired operations.

2.0 TRAID PRIMER

This section is designed to acquaint the reader with the basic concepts of TRAID. It shows simple solutions to simple problems; a more complete description of TRAID's capabilities is presented in the next section.

2.1 FUNDAMENTALS

To illustrate the most fundamental aspects of the use of TRAID, let's choose a typical-but-easy problem and develop a complete solution for it. (Note that this problem is too easy to demonstrate the power of TRAID; it is intended only as a pedagogic device.)

As a sample problem, say we are given the positions and velocities of an object and an interceptor, and we want a present-estimate of time-to-closest-approach. Expressed in vector equations, we have

$$\frac{d}{d\tau} \left| (\vec{PO} + \vec{VO}\tau) - (\vec{PI} + \vec{VI}\tau) \right|^2 = 0$$

$$2(\vec{PO} - \vec{PI}) \cdot (\vec{VO} - \vec{VI}) + 2(\vec{VO} - \vec{VI})^2 \tau = 0$$

$$\tau = - \frac{(\vec{PO} - \vec{PI}) \cdot (\vec{VO} - \vec{VI})}{(\vec{VO} - \vec{VI})^2}$$

where τ is the time (from now) to closest approach and the (current) values of position and velocity for object and interceptor are $\vec{PO}$, $\vec{PI}$, $\vec{VO}$, and $\vec{VI}$.

Now, to translate from an algebraic expression to a FORTRAN program, we need to do three things:

define position and velocity vectors
calculate the vector differences
calculate the dot products

The first of these is ordinary FORTRAN lore. Let's declare 3-component vectors PO, PI, VO, and VI, and for the differences, DP and DV. The corresponding DIMENSION statement is shown below.

As to calculating the vector differences, TRAID offers a subroutine for just this purpose. Its name is SUBVEC, and it takes three arguments, each a 3-component vector -- SUBVEC subtracts the second vector from the first, and stores the difference in the third.

The easiest way to calculate dot-products is to use the DOT function, which takes two vectors as arguments and returns their dot-product as the function value.

The resulting code is:

```
DIMENSION PO(3), PI(3), VO(3), VI(3), DP(3), DV(3)
CALL SUBVEC (PO, PI, DP)
CALL SUBVEC (VO, VI, DV)
TIMTOCA = - DOT(DP,DV)/DOT(DV,DV)
```

So much for the calculations. We have yet to arrange for reading-in the position-velocity data and for printing TIMTOCA. TRAID provides subroutines for reading and printing data, and it is recommended that you use them exclusively.

The simplest input routine is IN1; it reads data from a standard-format card, one value per card. You CALL IN1 (X,N), and IN1 will read N cards, storing successive values in (the list) X. A more convenient routine for vector-data, however, is IN3, which reads three values from each data-card. These cards may have any descriptive text punched in columns 1-40, and should have three values punched in col. 41-50, 51-60, and 61-70 in F10.0* format.

A simple printout routine is RITEF; its use is:

```
CALL RITEF (10HNAME/DESCR, VALUES, N)
```

When called in this fashion RITEF will print the 10-character NAME/DESCR, followed by N VALUES, in F-format. If the numbers in VALUES are integers, you should call RITEI, which prints I-format.

---

* If you punch the decimal, of course, you may place the value anywhere in the field.

A more flexible printout routine is subroutine WRITIT. When using WRITIT, you compose any string of BCD characters (like an ordinary sentence) which contains short strings of zeros where you want successive VALUEs to be inserted. Then you CALL WRITIT(VALUE1, INDENT, nHBCDSTRING-), where INDENT controls the left-to-right position of the printed line. You continue to CALL WRITIT (VALUE 2) and CALL WRITIT (VALUE 3), on until you have provided all the VALUEs needed to fill the zero-strings -- at this time the line is printed. An example of WRITIT's use appears in Sec. 2.5.

Two subroutines are provided for skipping lines and pages on the printout. They are CALL LSKIP(N) to skip N lines and CALL HEAD (N) to write a page-heading and skip N lines on a new page.

There is one more requirement -- and this is important -- all programs which use TRAID must make a special call of HEAD at the beginning. This is of the form: CALL HEAD (nHPROGRMNAME PROGRAMDESCRIPTION-). When called in this way, HEAD reads a data-card and performs several "supervisory" tasks:

- Establishes time and page limits (to be enforced on later calls to HEAD)
- Establishes physical constants and unit-scaling factors
- Establishes control parameters for the computation -- e.g., choice of integration methods
- Prints a title page, including the name and description of the program, and the number and description of the problem now being solved
- Assembles a page-heading, to be printed at the top of each page of printout.

Shown on the next five pages are:

- A program to solve our closest-approach problem, complete with input and output and special initial call to HEAD, and arranged to allow multiple cases to be computed. (The call of PREDATA causes the images of the data-cards to be printed.)
- A typical data-deck setup
- The resulting printout.

```
                      PROGRAM PRIMER1(INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT)
000003                DIMENSION PO(3),PI(3),VO(3),VI(3),DP(3),DV(3)
000003                CALL PREDATA
000004                CALL HEAD(34HPRIMER1 - FINDS CLOSEST APPROACH -)
000006                CALL HEAD(3)
000010              1 CALL IN3(PO,3)
000012                CALL IN3(VO,3)
000014                CALL IN3(PI,3)
000016                CALL IN3(VI,3)
000020                CALL SUBVEC(PO,PI,DP)
000023                CALL SUBVEC(VO,VI,DV)
000026                TIMTOCA=-DOT(DP,DV)/DOT(DV,DV)
000035                CALL RITEF(4HTIME,TIMTOCA,1)
000040                CALL LSKIP(5)
000042                GO TO 1
000043                END
```

IMAGES OF DATA-CARDS

```
          1         11        21        31        41        51        61        71
CARD      ....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....
    1  P1 DEMONSTRATION FOR PRIMER 1  -
    2  OBJECT POSITION                    0.        10000.    6000.
    3  OBJECT VELOCITY                    0.        -4000.    -2000.
    4  INTERCEPTOR POSITION             250.        8500.     4500.
    5  INTERCEPTOR VELOCITY            -100.        1500.     800.
    6  OBJECT POSITION                    0.        8000.     4000.
    7  OBJECT VELOCITY                    0.        -3900.    -2100.
    8  INTERCEPTOR POSITION             200.        9250.     4900.
    9  INTERCEPTOR VELOCITY             -80.        1300.     400.
```

PRIMER1 - FINDS CLOSEST APPROACH -

RUN P1 09/03/69 TIME 20.05.38.

DEMONSTRATION FOR PRIMER 1 -

COMPUTATION UNITS	-	METRIC
INTEGRATION METHOD	-	RECTANGULAR
RUNNING TIME LIMIT	-	999 MINUTES
OUTPUT LIMIT	-	999 PAGES

TRAID SYSTEM
GENERAL RESEARCH CORP.
5383 HOLLISTER AVENUE
GOLETA, CALIFORNIA

[illegible] INTEGRATION FOR PRIMER 1 =

[illegible]	0.0000	10000.0000	5000.0000
[illegible]	0.0000	-6000.0000	-2000.0000
[illegible]	250.0000	8500.0000	4500.0000
[illegible]	-100.0000	1500.0000	800.0000

[illegible] TIME = .2593937

OBJECT POSITION	0.0000	8000.0000	4000.0000
OBJECT VELOCITY	0.0000	-3900.0000	-2100.0000
INTERCEPTOR POSITION	200.0000	9250.0000	4900.0000
INTERCEPTOR VELOCITY	-80.0000	1300.0000	800.0000

[illegible] TIME = -.2596084

```
PRIMER1 - RUN P1          DEMONSTRATION FOR PRIMER 1  -

                                - END OF RUN -

                          EXECUTION TIME  0 HRS  00 MIN  00 SEC
```

2.2 STATE VECTORS

Trajectory computations in TRAID make use of (augmented) state vectors -- 10-component vector defined as follows:

STATE (1) = time in seconds

STATE (2-4) = position (x-y-z) in meters*

STATE (5-7) = velocity ($\dot{x}$-$\dot{y}$-$\dot{z}$) in meters*/second

STATE (8-10) = acceleration ($\ddot{x}$-$\ddot{y}$-$\ddot{z}$) in meters*/second^2

STATE(1) is the time when the body has the given position, velocity, and acceleration.

These are measured in a coordinate system which is determined by the control parameter KOORD -- see the figure on the next page. For any TRAID computation that depends on the altitude, KOORD must be set to the appropriate value: 0, 1, or 2. Normally, KOORD is set to zero (by the special call of HEAD); to change it to some other value N, you simply CALL SETKORD (N).

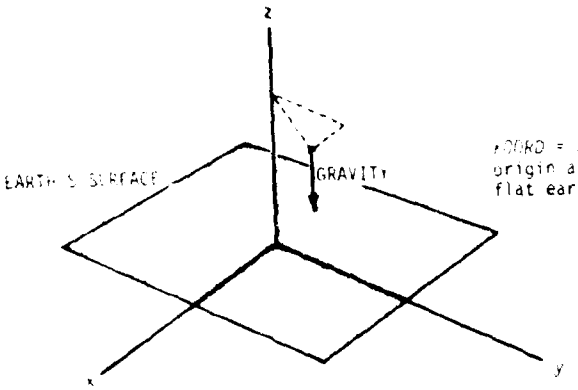
There are TRAID routines designed especially for the input and output of state vectors. They allow a variety of (external) coordinate systems; they allow the use of BCD names identifying the vectors; and they will handle all or specified parts of the state vectors.

The state-vector input routine is STIN; it does the following:

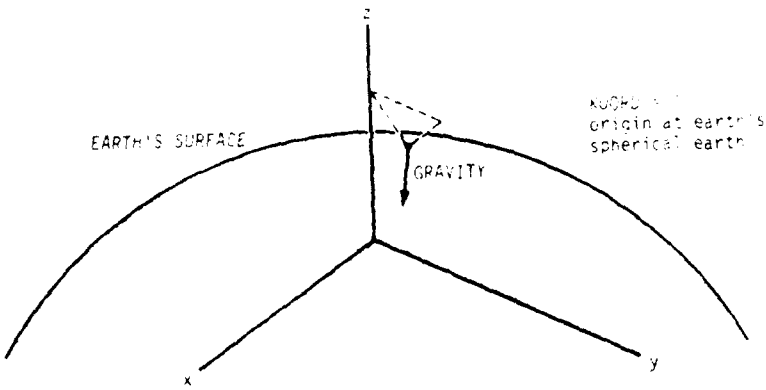
- Reads data-cards, extracting any or all of:
  - BCD names of vectors
  - time-component
  - position-components
  - velocity-components
- Transforms coordinates if necessary
- Stores the vector in an input array
- Prints the contents of the data-cards

---

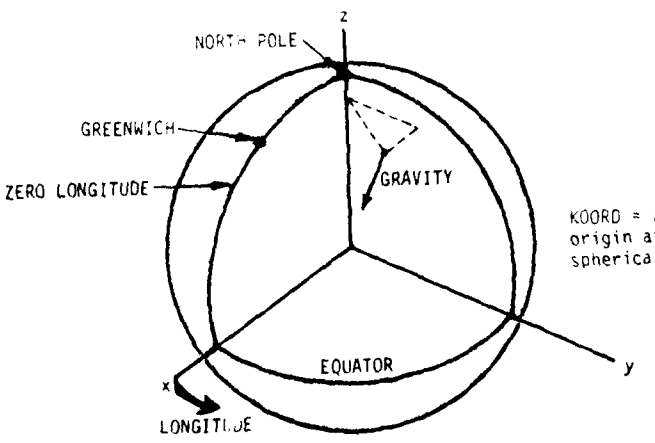
*Instead of meters, these will be in feet if KUNITS has been set, on the HEAD data card.



KOORD = 0
origin at earth's surface
flat earth



KOORD = 1
origin at earth's surface
spherical earth



KOORD = 2
origin at center of earth
spherical earth

PRECEDING PAGE BLANK-NOT FILMED

The state-vector printout routine is STOUT--it transforms coordinates if necessary and prints BCD names and/or time and/or position and/or velocity and/or acceleration

The arguments of SIIN and STOUT are the same:

- TITLE is a BCD title to be printed as indentification
- KFORM is a control code (of 6 digits or less) which specifies the external coordinate system. For example, KFORM = 110 means position plus velocity in x-y-z coordinates and KFORM = 400 means position only in alt-long-lat coordinates.
- NAMES is the location of an array of BCD vector-names. If the thousands' digit of KFORM is zero, a dummy argument may be used for NAMES.
- STATES is the location of the state-vector array
- NST is the number of state vectors to be input or output. If NST is negative, title and column-headings are not printed. If NST is zero, the title and column-headings are printed but no vectors are processed.

In order to illustrate the use of state vectors in TRAID, let's look at a noisy-radar-observation problem. Let's provide, as input data, a radar location, a target-object position and velocity, and a "state vector" of radar errors. Then we'll print a set of ten noisy position-velocity observations. These observations are produced by subroutine RADAR, as shown in the program listing on the next page.

```
000001          PROGRAM PRIMER2(INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT)
000003          DIMENSION SR(10),SO(10),DR(10),RO(10)
000004          CALL PREDATA
000006          CALL HEAD(29HPRIMER2 - SHOWS RADAR NOISE -)
000010        1 CALL HEAD(1)
000014          CALL STIN(12HRADAR LOCN =,100,DUMMY,SR,1)
000020          CALL STIN(13HOBJECT TRAJ =,110,DUMMY,SO,1)
000024          CALL STIN(14HRADAR ERRORS =,330,DUMMY,DR,1)
000030          CALL LSKIP(3)
000032          CALL STOUT(25HNOISY OBSERVATIONS =,110,DUMMY,DUMMY,0)
000034          DO 2 K=1,10
000037          CALL RADAR(SO,SR,DR,RO,1)
000045        2 CALL STOUT(DUMMY,110,DUMMY,RO,-1)
000046          GO TO 1
                END
```

PRECEDING PAGE BLANK-NOT FILMED

IMAGES OF DATA-CARDS

```
          1         11        21        31        41        51        61        71
CARD      *.........*.........*.........*.........*.........*.........*.........*.....
   1  P2 DEMONSTRATION FOR PRIMER 2 -
   2  RADAR UPRANGE OF TARGET                      0.        50.       0.
   3  OBJECT ON ICBM TRAJECTORY                    0.        100.      20.
   4     SPEED = 7071 M/SEC                        0.        -7000.    -1000.
   5  POSITION ERRORS (KM,DEG,DEG)                1.        1.        1.
   6  VELOCITY ERRORS (M,DEG,DEG)/SEC             2.        3.        3.
```

PRIMER2 - RUN P2 DEMONSTRATION FOR PRIMER 2 -

RADAR LOCN -

POSITION COORDINATES		
X KM	Y KM	Z KM
0.000	50.000	0.000

OBJECT TRAJ -

POSITION COORDINATES			VELOCITY COORDINATES		
X KM	Y KM	Z KM	X M/SEC	Y M/SEC	Z M/SEC
0.000	100.000	20.000	0.000	-7000.000	-1000.000

RADAR ERRORS -

POSITION COORDINATES			VELOCITY COORDINATES		
RANGE KM	AZIMUTH DEG	ELEVATION DEG	MAGNITUDE M/SEC	AZIMUTH DEG	ELEVATION DEG
1.000	1.000	1.000	2.000	3.000	3.000

NOISY OBSERVATIONS -

POSITION COORDINATES			VELOCITY COORDINATES		
X KM	Y KM	Z KM	X M/SEC	Y M/SEC	Z M/SEC
.596	99.108	20.306	-63.255	-6979.528	-1102.766
-.625	100.544	19.585	46.291	-7017.492	-907.304
.307	99.928	19.734	41.178	-7003.607	-972.690
-.921	100.936	20.148	127.756	-7015.945	-942.494
-.525	99.896	20.398	113.953	-6990.361	-1058.557
-1.140	99.405	20.392	172.298	-6984.509	-1087.290
-.028	99.957	20.541	-70.946	-6993.865	-1058.498
-.773	100.376	19.955	5.999	-7008.917	-962.388
1.396	101.221	19.380	-123.494	-7029.517	-839.772
-1.324	99.308	18.606	291.621	-7005.025	-900.660

2.3 TRAJECTORY INTEGRATION

TRAID integrates trajectories with the following features:

1. thrust, mass, lift, drag, maneuver limits, and delayed response, for up to 3 stages
2. guidance commands, either dynamically computed or pre-selected
3. termination-rule defined by the user

For example, you might (1) define a vehicle with lift and drag only, (2) input some preselected maneuver commands, and (3) ask for the vehicle's trajectory to be integrated until it reaches some specified altitude, or some specified speed, or some specified heading, or some other criterion.

Or you might (1) describe a rocket with three stages of thrust, lift, and drag, (2) provide a proportional-navigation routine which computes guidance commands toward a target object, and (3) request integration of the trajectory for a specified time interval, or other criterion.

In both of the above examples, a single call of one subroutine will produce the requested trajectory, as a collection of successive state vectors (the first vector represents the start of integration, and the last represents the state where the termination-condition is attained).

The rocket description is contained in an array usually named C and dimensioned 3 (stages) by 16 (descriptors). These descriptors, again, are parameters such as thrust, mass, lift, drag, response, etc. for each stage. Another array, usually named CP and singly-dimensioned, contains staging schedule and guidance information.

There is a TRAID subroutine (FLIN) designed to read information from data-cards, and store it in the C and CP arrays. FLIN takes 4 arguments:

- TITLE, a BCD title, ending on a dash
- MODE, a three-digit number
    a. first digit controls reading of thrust/mass data
    b. second digit controls reading of drag/response data
    c. third digit controls reading of preselected maneuver commands
- C, CP } rocket-descriptor arrays where data is to be stored

For data of types (a) and (b) above, a more complete description appears under subroutine FLIN in Sec. IV. In the case of type (c) data, for preselected maneuvers, you punch on cards the capitalized entities in this command: "pull ANC g's (in a direction normal to velocity and described by PSI and PSIDOT) until parameter NAME reaches a value of VAL." Typical examples are:

ANC	PSI	PSIDOT	NAME	VAL	Interpretation
25	+90°	0	VEL.EL.	-45°	"Dive at 25 g's until you're heading at 45°."
10	0	60 deg/s	TIME INT	6.0	"Turn a 10-g barrel roll in 6 seconds."

Any number of the maneuver commands may be stacked -- they will be performed sequentially.

The TRAID routine which carries out the trajectory integration is subroutine FLIGHT. When you call FLIGHT, you must provide:

- C and CP arrays to describe the rocket and its guidance, and the STATES array to be filled with subsequent state vectors.
- the names of three other subprograms:
    - One to calculate commanded acceleration -- e.g. FLIER
    - One to calculate actual acceleration -- e.g. FLAC or RV
    - One to interpret the termination condition -- e.g. FOAL

- integration control parameters: a time step, a parameter name and value for termination, and NOUT (which causes a vector to be inserted into the STATES array every NOUT-th time step).
- NST, which serves two purposes: it advises FLIGHT how many vectors you expect will be stored in STATES, and it tells your program how many vectors actually were stored there.

To illustrate the above points, the program on the next page calls FLIN to input a rocket description and again for a set of maneuver commands -- then it calls FLIGHT to integrate a trajectory and STOUT to print it. Notice these items:

- The dimension of CP must be 20+5*N to allow N maneuvers.
- FLAC, FLIER, and FOAL must appear in an EXTERNAL statement.
- An initial position and velocity must be set up in STATES.
- Before calling FLIGHT, NST is set to the size of STATES.
- After returning from FLIGHT, NST contains the number of vectors.
- FLIGHT is asked to integrate in 0.1-second steps until the altitude reaches 40 km, and to insert every 10th vector in the STATES array.

```
                PROGRAM PRIMER3(INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT)
000033          DIMENSION C(3,10),CP(3,3),STATES(10,100)
000043          EXTERNAL FLAC,FLIER,FOAL
000053          CALL PREDATA
000056          CALL HEAD(26HPRIMER  - FLIES A ROCKET =)
000064          CALL HEAD(1)
000070          CALL RUN
000071          CALL FLIP(11HINT PERF  =,230,C,CP)
000076          CALL HEAD(1)
000104          CALL STIM(19HSTARTING POSITION =,110,DUMMY,STATES,1)
000122          CALL FLIP(11HCOMMAND   =,01,C,CP)
000126          NST=100
000132          CALL FLIGHT(FLAC,FLIER,STATES,C,CP,0.1,FOAL,.0000,,4HALTITUDE,
     *                10,NST)
000161          CALL STOR(12HTRAJECTORY =,1130,DUMMY,STATES,NST)
000175          GO TO 1
000204          END
```

PRECEDING PAGE BLANK-NOT FILMED

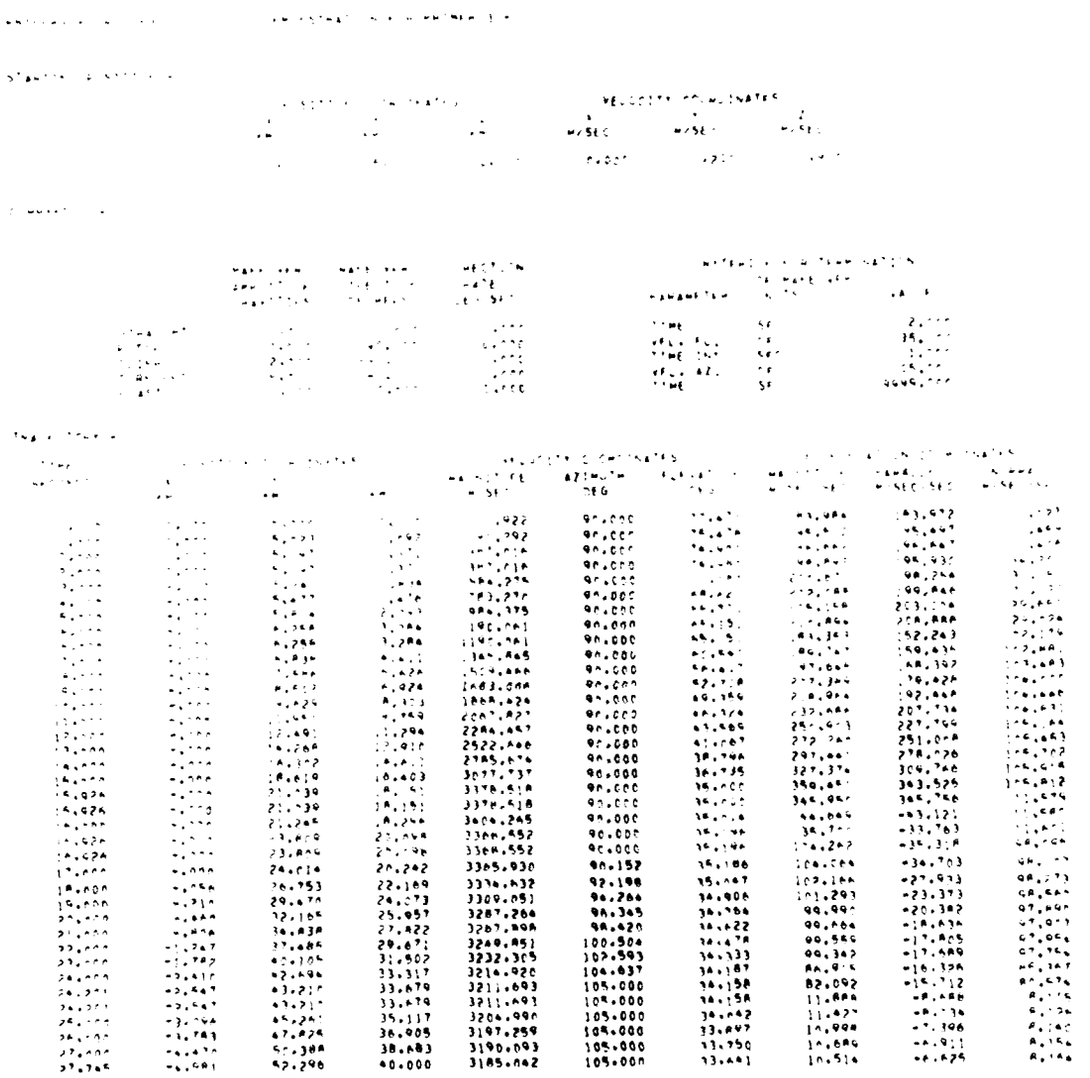
IMAGES OF DATA-CARDS

```
      CARD  ....1....+....2....+....3....+....4....+....5....+....6....+....7....
        1   P3 DEMONSTRATION FOR PROBLEM 3 =
        2   VACUUM THRUST, LB                          100000.    80000.     0
        3   NOZZLE AREA, SQ FT                            144.       72.     0
        4   INITIAL MASS, LB                             8000.     4000.     0
        5   FINAL  MASS, LB                              5000.     2000.     0
        6   BURN TIME, SEC                                 4.       10.     0
        7   REF AREA, SQ FT                                4.      1.5      0
        8   MACH .8 DRAG COEFF                            .5        .5      0
        9   MACH 1+ DRAG COEFF                            1.5       1.      0
        A   MACH 3  DRAG COEFF                            1.        .7      0

      CARD  ....1....+....2....+....3....+....4....+....5....+....6....+....7....
        11  MACH 10 DRAG COEFF                            .5        .4      0
        12  LIFT LIMIT (STRUCTURAL), G-S                 2.5       10.     0
        13  ATTACK LIMIT (STABILITY), DEG                2.5        5.     0
        14  CN/A COEFF, 1/DEG                             1.        1.     0
        15  CM/A/A COEFF, 1/DEG SQ                       0.1       0.2     0
        1A  ROCKET LAUNCH LOCATION          0             0.        5.     0.
        17  LAUNCH DIRECTION                              0.        .2     .9
        18  STRAIGHT   0.        0.        0.          TIME       SEC      2.
        19  PITCH      10.       90.       0.          VEL. EL.   DEG      85.
        20  CLIMB      2.        -90.      0.          TIME INT   SEC      1.

      CARD  ....1....+....2....+....3....+....4....+....5....+....6....+....7....
        21  TURNLEFT   15.       180.      0.          VEL. AZ.   DEG      105.
        22  COAST      0.        0.        0.          TIME       SEC      9999.
        23  END DATA
```

PRIMER3 - RUN P3 DEMONSTRATION FOR PRIMER 3 -

INT PERF -

	STAGE ONE	STAGE TWO	STAGE THREE
VACUUM THRUST, LB	140000.000	80000.000	0.000
NOZZLE AREA, SQ FT	144.000	72.000	0.000
INITIAL MASS, LB	8000.000	4000.000	0.000
FINAL MASS, LB	5000.000	2000.000	0.000
BURN TIME, SEC	6.000	10.000	0.000
REF AREA, SQ FT	4.000	1.400	0.000
MACH .8 DRAG COEFF	.400	.600	0.000
MACH 1.2 DRAG COEFF	1.500	1.000	0.000
MACH 3 DRAG COEFF	1.000	.700	0.000
MACH 10 DRAG COEFF	.600	.400	0.000
JET LIMIT (STRUCT FAIL + G-S	2.500	10.000	0.000
ATTACK LIMIT (STAB, CTRL), DEG	2.400	5.000	0.000
CN/A COEFF, 1/DEG	1.000	1.000	0.000
CA/A/A COEFF, 1/DEG SQ	.100	.200	0.000



2.4 ORBITS

TRAID offers an alternate representation of position-velocity -- namely Keplerian orbits. It is true that most of the geometric calculations and all of the trajectory integration take place in the rectangular state-vector coordinates. Even so, there are some problems and parts of problems which can be solved more easily by using orbits; and in many problems the easiest solution involves both state vectors and orbits, and frequent interchanges between the two.

TRAID uses an orbital-element set to describe a Keplerian orbit. By convention, this is a 10-component vector; it can describe either an elliptical or hyperbolic orbit.

The subroutine which reads and/or prints orbital-element vectors is subroutine ORIO. It takes 5 arguments:

- A BCD title
- A 3-digit control code. If the first digit = 0, ORIO prints but does not read data-cards. If it = 1, ORIO does both. If the second and third digits = 0, the 3rd argument may be a dummy.
- A list of names (may be dummy)
- Location where orbital elements are stored
- Number of orbital-element sets to be input/output

The typical uses of orbits fall into two categories: transforming from state vector(s) into orbit, and transforming back from an orbit to a state-vector.

For the first of these transformations, TRAID offers two options. Subroutine ORB1 creates an orbital-element vector from the (1) time, (2) position, and (3) velocity in a state vector.

The second option is ORB2, which passes an orbit through the positions of two state vectors (using the time at one of them) and satisfies one other requirement:

Flight-time is a specified value

or Flight-time excess above minimum-energy is a specified value

or Speed at one of the positions is a specified value

or Velocity-elevation at one of the positions is a specified value

To "reenter" from orbit into state-vector form, you call ORBP with one extra parameter. You may choose between:

true anomaly	(MODE = 0)
radius (increasing)	(MODE = 1)
radius (decreasing)	(MODE = 2)
time	(MODE = 3)
altitude (increasing)	(MODE = 4)
altitude (decreasing)	(MODE = 5)

If you want only the time when one of the above parameters reaches a given value, you may use the function ORBTIME.

Some illustrative uses of the orbit routines are shown in the example in the next section.

2.5 EXAMPLE

On the following pages is a sample program which exercises most of the features discussed throughout Secs. 2.1-2.4. This program finds the closest approach between an ICBM-type object and an interceptor which is launched when the object descends to some specified altitude, flies a preselected maneuver until it reaches 50 km altitude, and then goes "into orbit." During the search for closest approach, both bodies are described by orbital elements -- these must be converted to state vectors for each trial solution.

Notice the following items in the sample program:

- There are many 10-vectors; ST is dimensioned 10 × 20, providing room for 20 successive states; CP is dimensioned 45, allowing 5 maneuver phases.
- FLAC, FLIER, FOAL must be declared in an EXTERNAL statement.

- The use of Keplerian orbits, in conjunction with atmospheric flight, demands that KOORD be set to 2.
- The object's trajectory is to be determined by launch and impact-points, and a MODE-VALUE pair for ORB2.
- The time for interceptor launch is determined by an input object altitude. The object's orbit time is reset so that this altitude is reached at time = zero.
- The interceptor's performance, its preselected maneuver, and its launch site are read in, and its clock ST (1,1) is set to the launch time = zero.
- Before calling FLIGHT, MAX is set to 20, the size of the ST array; FLIGHT inserts every 50th vector into ST and stops at 50 km altitude with MAX now containing the actual number of vectors in ST. The trajectory is printed and a set of orbital elements is produced.
- In searching for closest approach, we start at the time the interceptor reached 50 km and we will search until we find positions within 1 second (TCLOSE) of closest approach.
- For each iteration, we obtain state vectors (S0 and S1) from each orbit, compute the differences DP and DV, and find the current TIMTOCA. If this is greater than TCLOSE, we reset TIMTRY and repeat.
- When we find a TIMTOCA which is less than TCLOSE, we advance the positions linearly for that time, find the position difference, and print the time and miss-distance.
- There are three routines (which appear after statement 20) which have not been described yet:
    - CALL VECLIN (a,A,b,B,C) produces a 3-vector C which is a linear combination of the 3-vectors A and B -- i.e., $\vec{C} = a\vec{A} + b\vec{B}$
    - X = XMAG (A) returns the length of 3-vector A.
    - CALL CEASE (12HBCDMESSAGE-) prints a BCD message, then prints a run-ending page, and stops the program.

```
      PROGRAM EXAMPLE4 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
000003      DIMENSION ST(10,20),S(10),ORA(10),ORB(10),C(3,10),CP(6),S1(10),
     *   S2(10),V(3),CP(3)
000023      EXTERNAL FLAC,FLIER,FCL
C
000031      CALL PRELDATA
000034      CALL MFAC(3,'EXAMPLE4 - 3 DIMENSIONAL TRAJECTORY')
000046      CALL MFAC(1)
000051      CALL SETXPRC(2)
C
000057      CALL STIN(20,'OBJECT LAUNCH SITE =',000000,A,S,1)
000074      CALL STIN(20,'OBJECT IMPACT SITE =',000000,A,ST,1)
000117      CALL LSKIP(3)
000124      CALL IN1(MODE,-1)
000132      CALL IN1(VALUE,1)
000140      CALL ORH2(MODE,VALUE,N,ST,'RH')
000164      CALL ORIC(1,'OBJECT ORBIT =',000,A,'ORB',1)
000220      CALL LSKIP(3)
000225      CALL IN1(ALTFIRE,1)
000233      TIMEFIRE=ORBITIME(1,A,TGT,F,ORB)
000250      ORB(7) = ORB(1)-TIMEFIRE
C
000261      CALL MFAC(1)
000263      CALL FLIN(25,'INTERCEPTOR PERFORMANCE =',231,C,CP)
000305      CALL STIN(25,'INTERCEPTOR LAUNCH SITE =',000030,A,ST,1)
000342      CALL LSKIP(3)
000346      CALL IN1(DT,1)
000356      ST(1,1)=0.
000367      MAX=20
000370      CALL FLIGHT(FLAC,FLIER,ST,C,CP,DT,FOAL,50,E3,'HALTITUDE',50,MAX)
000103      CALL ORH1(ST(1,MAX),'ORA1')
000127      CALL STMEP(ST,ST,MAX,1)
000132      CALL MFAC(5)
000134      CALL STOUT(24,'INTERCEPTOR TRAJECTORY =',0014,30,A,ST,MAX)
C
000220      TCLOSE=1.
000222      TIMTRY=ST(1,MAX)
000225   10 CALL ORBP(3,TIMTRY,ORB,S0)
000330      CALL ORBP(3,TIMTRY,ORB,S1)
000333      CALL SUBVEC(S0(2),S1(2),DP)
000343      CALL SUBVEC(S0(5),S1(5),DV)
000361      TIMTOCA=-DOT(DP,DV)/DOT(DV,DV)
000450      IF(ABS(TIMTOCA).LT.TCLOSE) GO TO 20
000453      TIMTRY=TIMTRY+TIMTOCA
000456      GO TO 10
C
000465   20 CALL VECLIN(1.,S0(2),TIMTOCA,S0(5),S0(2))
000541      CALL VECLIN(1.,S1(2),TIMTOCA,S1(5),S1(2))
000614      CALL SUBVEC(S0(2),S1(2),DP)
000700      DMISS=MAG(DP)*.001
000774      CALL LSKIP(3)
000775      CALL WRITIT('TIMTRY+TIMTOCA,0,'SHAT TIME 0000.00, MISS DISTANCE IS
     *00,000 KM =')
000202      CALL WRITIT(DMISS)
000206      CALL CEASE(13,'NORMAL EXIT =')
00020A      END
```

```
INPUT DATA LISTING

      1         11        21        31        41        51        61        71
CARD ....*....|....*....|....*....|....*....|....*....|....*....|....*....|....*....|
   1  P& DEMONSTRATION ECHO PRIMARY &P
   2  ICRM LAUNCH SITE                    0         0.        0.        75.
   3  ICRM TARGET SITE                    0         50.       90.       0.
   4  MODE ECHO CRM2                              4.
   5  VALUE ECHO CRM2                             33.       DEG
   6  TERM ALT WHEN INTERCEPTOR IS LAUNCHED       200.      KM
   7  VACUUM THRUST, LM                           100000.   80000.    0
   8  NOZZLE AREA, SQ FT                          144.      72.       0
   9  INITIAL MASS, LM                            8000.     4000.     0
  10  FINAL  MASS, LM                             4000.     2000.     0

      1         11        21        31        41        51        61        71
CARD ....*....|....*....|....*....|....*....|....*....|....*....|....*....|....*....|
  11  G-SW TIME, SEC                              6.        10.       0
  12  REF AREA, SQ FT                             4.        1.5       0
  13  MACH .5 CMAG COEFF                          .4        .5        0
  14  MACH 1.0 CMAG COEFF                         1.5       1.        0
  15  MACH 1.3 CMAG COEFF                         1.        .7        0
  16  MACH 10 CMAG COEFF                          .5        .4        0
  17  LIFT LIMIT (STRUCTURAL), GS                 2.5       10.       0
  18  ATTACK LIMIT (STABILITY), DEG               2.4       5.        0
  19  CNA1, PER, RADS                             1.        1.        0
  20  CNA2A COEFF, RADS**-2                       -.1       0.2       0

      1         11        21        31        41        51        61        71
CARD ....*....|....*....|....*....|....*....|....*....|....*....|....*....|....*....|
  21  STRATCNT  0.        0.        0.        TIME      SEC       2.
  22  PITCH     10.       90.       0.        VEL, EL,  DEG       35.
  23  CLIMB     2.        -90.      0.        TIME INT  SEC       1.
  24  T WALKOFF 15.       180.      0.        VEL, AZ,  DEG       105.
  25  COAST     0.        0.        0.        TIME      SEC       9999.
  26  ***DATA
  27  INTERCEPTOR FARM LOCATION       0         0.        90.05     -0.4
  28  INTERCEPT W, LAUNCH VECTOR                  1.        90.       80.
  29  INTEGRATION TIME STEP                       .1
```

PRIMER6 - RUN P6 DEMONSTRATION FOR PRIMER 6 -

- ORIGIN IS AT CENTER OF ROUND EARTH (KOORD=2) -

OBJECT LAUNCH SITE -

POSITION COORDINATES		
ALTITUDE KM	LONGITUDE DEG	LATITUDE DEG
0.000	0.000	75.000

OBJECT IMPACT SITE -

POSITION COORDINATES		
ALTITUDE KM	LONGITUDE DEG	LATITUDE DEG
50.000	90.000	0.000

MODE FOR ORB2 4.0000*
VALUE FOR ORB2 33.0000 DEG (.5749587 AFTER SCALING)

OBJECT ORBIT -

LONG. OF ASCEND. NODE DEG	INCLINATION DEG	ARG. OF PERIGEE DEG	SEMI-LATUS RECTUM KM	ECCENTRICITY	TIME AT PERIGEE SEC	PERIOD SEC	SEMI-MAJOR AXIS KM
-90.000	75.000	315.345	3883.455	.550104	-857.897	6195.416	5622.101

ICBM ALT WHEN INTERCEPTOR IS LAUNCHED 200.0000 KM (200000.0 AFTER SCALING)

PRIMER6 - RUN P6 DEMONSTRATION FOR PRIMER 6 -

INTERCEPTOR PERFORMANCE -

	STAGE ONE	STAGE TWO	STAGE THREE
VACUUM THRUST, LB	140000.000	80000.000	0.000
NOZZLE AREA, SQ IN	144.000	72.000	0.000
INITIAL MASS, LB	8000.000	4000.000	0.000
FINAL MASS, LB	5000.000	2000.000	0.000
BURN TIME, SEC	6.000	10.000	0.000
REF AREA, SQ FT	4.000	1.400	0.000
MACH .5 DRAG COEFF	.500	.400	0.000
MACH 1. DRAG COEFF	1.500	1.000	0.000
MACH 3. DRAG COEFF	1.000	.700	0.000
MACH 10 DRAG COEFF	.500	.400	0.000
LIFT LIMIT (STRUCTURAL), G'S	2.500	10.000	0.000
ATTACK LIMIT (STABILITY), DEG	2.500	5.000	0.000
CN/A COEFF, 1/DEG	1.000	1.000	0.000
CM/A/A COEFF, 1/DEG S.	.100	.200	0.000

	MANEUVER AMPLITUDE GRAVITIES	MANEUVER DIRECTION DEGREES	DIRECTION RATE DEG/SEC	CRITERION FOR TERMINATION OF MANEUVER PARAMETER	UNITS	VALUE
STRAIGHT	0.000	0.000	0.000	TIME	SEC	2.000
PITCH	10.000	90.000	0.000	VEL. EL.	DEG	35.000
CLIMB	2.000	+90.000	0.000	TIME INT	SEC	1.000
TURNLEFT	15.000	180.000	0.000	VEL. AZ.	DEG	105.000
COAST	0.000	0.000	0.000	TIME	SEC	9999.000

INTERCEPTOR LAUNCH SITE -

POSITION COORDINATES ALTITUDE KM	LONGITUDE DEG	LATITUDE DEG	VELOCITY COORDINATES MAGNITUDE M/SEC	AZIMUTH DEG	ELEVATION DEG
0.000	90.050	+.400	1.000	90.000	90.000

INTEGRATION TIME STEP .1000

PRIMENA - RUN P4 DEMONSTRATION FOR PRIMEN 6 -

INTERCEPTOR TRAJECTORY -

TIME SECONDS	POSITION COORDINATES ALTITUDE KM	LONGITUDE DEG	LATITUDE DEG	VELOCITY COORDINATES MAGNITUDE M/SEC	AZIMUTH DEG	ELEVATION DEG
0.000	-.000	90. 50	-.480	1.000	90.000	90.000
2.000	.375	90.050	-.599	384.881	90.000	78.008
2.000	.375	90.050	-.599	384.881	90.000	78.008
4.000	2.332	90.050	-.594	984.111	90.000	69.862
6.000	5.367	90.050	-.590	1190.066	90.000	60.362
8.000	5.367	90.050	-.590	1190.066	90.000	60.362
10.000	8.550	90.050	-.563	1069.819	90.000	52.456
15.000	17.187	90.050	-.486	3090.966	90.000	40.187
17.000	23.100	90.050	-.417	3360.053	90.000	35.000
17.000	23.100	90.050	-.417	3360.053	90.000	35.000
18.000	25.828	90.050	-.393	3339.175	90.000	34.224
18.000	25.828	90.050	-.393	3339.175	90.000	34.224
20.000	27.124	90.049	-.366	3316.703	92.750	35.099
25.000	36.607	90.036	-.267	3233.602	101.497	36.513
30.000	45.480	90.007	-.132	3187.707	104.758	33.916
32.594	90.000	89.991	-.074	3170.148	105.620	31.607

AT TIME 37.39. MISS DISTANCE IS 5.983 KM -

NOMINAL EXIT -

3.0 TRAID GENERALITIES

The remarks in this section are intended to fill the gap between the Sec. 2.0 Primer, which was deliberately simplified and incomplete, and the Sec. 4.0 Subroutine Listings, which do not present a coherent view of TRAID. To state it another way, if Sec. 4.0 describes the building block routines making up the TRAID family, this section provides the blueprint for assembling them and the mortar which holds them together.

On the following pages, arranged by subject matter, are described TRAID's operating principles, some comparative evaluations, and hints for use.

3.1 PROGRAM CONTROL

Operating principles are:

- Execution-time-limit and printout-page-limit may be input via data-card; exceeding either limit causes run termination and (normal) exit to the monitor.
- Endfile marks on the input cause run termination and (normal) exit to the monitor -- this feature may be overridden if you wish to continue computing after an EOF. (See function CHEKFIL.)
- Error conditions arising during TRAID calculations -- such as failure to converge -- cause run termination and (normal) exit to the monitor; this feature may be overridden (if, for example, you wish to abandon the current case and proceed to read data for the next case) -- see function MISTAKE.

COMPKG. This is a main program which reads FORTRAN cards from the input file, inserts COMMON packages among them as needed, and writes the FORTRAN-plus-COMMON on a file named MERGED. See Sec. 5.1 for typical deck setup using a COMPKG 'control' card.

PREDATA/MORDATA. If you want to have all your data cards printed at the beginning of your run, call PREDATA. When using PREDATA, you must call it before any other TRAID input routine, and you must call it only once. Normally, your main program card will specify TAPE5=INPUT; if using PREDATA, this should be simply TAPE5.

If you want to be able to edit your data-cards (in order to run succesive cases), you may call MØRDATA. This routine writes an edited set of data-cards on TAPE5; editing may be accomplished by matching columns 1-20 of an old data card, or by more explicit requests to insert or delete specified numbers of cards. When using MØRDATA, your main program must provide a buffer for TAPE4. The complete rules for this data-editing routine may be found under PREDATA/MØRDATA in Sec. 4.0.

HEAD. The first TRAID routine called -- except for PREDATA -- should be CALL HEAD (nHPROGRMNAME PROGRAMDESCRIPTION-). At this call, HEAD reads a data card containing:

- A run number and run description, which together with PROGRMNAME, PROGRAMDESCRIPTION, the time, the date, and the page number, are written at the top of each page.
- Parameters which control the calculations--i.e., KSTINT and KUNITS (see Sec. 5.6).
- Optional execution-time-limit and printout-page-limit--these limits are enforced by HEAD when HEAD is called upon to skip a page on the output. If no value (zero or blank) is input for (either of) these limits, then HEAD will not cut your job off; this does not affect the monitor system's cutoffs.

It is on the occasion of this special call that HEAD sets up values for the physical constants used in the calculations, prints a title page, and sets KOORD to zero (see SETKORD, next).

SETKORD. TRAID's interpretation of coordinate systems depends on KOORD (see the figure in Sec. 2.2). There are in TRAID three areas of sensitivity to KOORD:

- Atmosphere models, embodied in DNSITY and GRAV
- Calculation of altitude, as in ALTF
- Interpretation of radar measurements in subroutine RADAR

Since the special call of HEAD causes KOORD to be set to zero, you have to CALL SETKORD (N) to give it a nonzero value.

SETKØRD also offers entry points convenient for (re)setting KSTINT, KUNITS, AND WBODY.

CHEKFIL. All TRAID input routines use function CHEKFIL to test for EOF on input. Normally, CHEKFIL will exit upon an EOF. To disable this automatic exit, CALL SETIFEF (1); to reenable, CALL SETIFEF (0). While the automatic exit is disabled, you ought to check for the occurrence of an EOF by looking at the number of endfiles, N = NUMFILS (DUMMY), after every call of an input routine.

WHEN. This routine prints elapsed computer time.

MISTAKE. All TRAID routines with error conditions call TRADERR, which prints an appropriate message and exits to the monitor. To disable this automatic exit, CALL SETIFER(1); to reenable it, CALL SETIFER(0). While the automatic exit is disabled, you ought to check for errors by consulting the number of errors, N = MISTAKE (DUMMY). [You might use this feature if you wish to proceed to a subsequent case even if there is an error in a previous case.]

Q8ERROR. This routine is called by TRADERR to write the trace of calling programs; it prints the name of, and relative address in, each subprogram in the chain back to the main program.

CEASE. Subroutine CEASE provides a convenient way to print a message, a run-termination page, and exit to the monitor -- all in one CALL statement. Example:

```
IF (NTRIES .GT. LIMIT) CALL CEASE (19HbITERATION FAILEDb-)
```

3.2 INPUT

Operating principles are:

- When data is read in, it is normally printed immediately in its original form. Thus, in normal operation, the printout shows exactly what data was used.
- If an endfile is found, normally an exit is made to the monitor -- this may be overridden, however.
- Data-cards are typically divided into 10-column fields, with BCD description on the left and numbers on the right, frequently beginning in column 31 or 41.
- All numbers are punched with decimal points: they will be converted to integers internally if appropriate.

- If 'OLD DATA' appears in column 1-8 on a data-card, printing and further reading are suppressed and no data is transferred into core (until the next call of an input routine).
- If 'END DATA' appears in column 1-8 on a data-card, this stops open-end reading (such as the reading of already-named variables in UGETIT and the reading of preselected maneuvers in FLIN).

- Some input routines accept a BCD title (which must end on a dash) to be printed above the table of printed data.
- Some input routines will read and return names from the left ends of data-cards -- typically, the NAMES are read from column 1-30, and they are stored in an array NAMES (5,LINES).
- Some input routines allow their printing to be suppressed.
- Some input and output routines use a SETUP array, which contains an output format and column-headings for up to ten 12-column print-fields. This SETUP array is assembled by subroutine OUTSET, and used by INCOL and OUTCOL. If you call OUTSET, you must provide a SETUP which is dimensioned (82).
- All input is read from logical unit 5 -- except for PREDATA, which reads from INPUT and writes on 5.

FLIN. Subroutine FLIN reads rocket-performance data and preselected maneuvers. Its operation is controlled by the 3-digit MODE, as follows:

- The first digit triggers the reading of either 0, 1, 5, or 7 cards describing thrust, mass, and burning schedule.
- The second digit, for values 0 through 4, triggers the reading of either 0, 2, 5, 9, or 11 data-cards describing axial and normal force coefficients, maneuver limits, and response. If this digit is a 5, FLIN reads 11 data-cards (as for 4), then calls INI to read an integer N from another data-card, and proceeds to read N more data-cards in the standard FLIN format. If this option is used, the C array should be dimensioned (3, 16 + N).
- The third digit triggers the open-end reading of preselected maneuvers. Reading continues until (1) a mark is found in

column 71-72 of the last card of actual data, or (2) 'END DATA' is found in column 1-8 of the card after the last card of actual data. To allow room for N phases of preselected maneuver, CP must be dimensioned (20 + 5N): if there are no maneuvers, the dimension may be only (13). Note that it is no longer necessary that the last maneuver have an unreachable goal.

If you call FLIN for a rocket to be launched at a time other than zero, you should set CP(7) = desired launch time, before calling FLIN.

ORIO. Subroutine ORIO reads orbital-element sets, which may be punched by hand (but rarely are), or which ORIO will punch for you if you call it with MODE = 2XX.

STIN. The TRAID routine which is designed to read state vectors is subroutine STIN. It will read time/position/velocity data in a variety of (external) coordinates, such as:

x - y - z
r - θ - ϕ
R - A - E
alt - long - lat

The choice of coordinates is implied by the 10's and 100's digits (KP and KV) in the second argument of the CALL statement -- but these may be overridden by KP and KV punched in columns 71-72 of the data-card. States read by STIN are always transformed to standard state-vector form--note that the converse of this rule does not apply to STOUT. STIN stops reading when it finds an 'END DATA' card, and resets LINES to the actual number read.

IN1, IN3. Subroutine IN1 reads N values, one value per card, and scales it by either (1) multiplier/divisor punched in column 51-70, or (2) unit-name punched in column 51-53. If the second argument N is positive, data is left in floating-point form; if negative, it is converted

to integer form and an asterisk is printed on the printout; if N is zero, one card is read and the BCD characters in column 41-50 are returned. Subroutine IN3 reads N values, three values per card, with no scaling capability. If the required number of values is satisfied in the 'middle' of a card, the values on the rest of that card are ignored.

INMV. Subroutine INMV reads from one data card a mode and a value --e.g., for use by ORB2 or ORBP. The value may be input in such units as DEG, NMI, etc., if the appropriate name is punched on the data card.

INDO. This is a routine which sets up and controls 'looping' variables. A call of INDO causes it to read data-cards which define the loop-variables -- the cards say "let X vary from A to B in a manner M." The manner M may be either linear or geometric progression or random; and the loops may be nested or concurrent. Later calls cause INDO to advance the variables to their next values. Note that (1) if all your loops are random, INDO cannot terminate and you must stop it, and (2) the loop variables are real (floating-point) variables and they must be stored contiguously in core.

UNAMIT/UGETIT. TRAID provides a routine for the purpose of changing the values of variables named on data-cards -- this allows you to delay the decision on variables-to-be-updated until data-punching time.

Subroutine UNAMIT has the following features:

- The 'changeable' variables must appear individually in calls to UNAMIT -- on these calls, UNAMIT saves the address of the variable and the name which appears in column 1-10 of the data-card, and of course, stores the value in the appropriate location.
- The changeable variables need not be contiguous, and the names on the data-cards need not match names which are built into the program -- they only need to be self-consistent.
- Data-values may be scaled by multiplier/divisor or by unit-name (as in IN1).
- Data-values are converted to integers if the name in column 1-10 begins with a letter I through N.
- After UNAMIT has built its table of changeable variables, you CALL UGETIT to cause data-cards to be read, their names matched, their values scaled and stored in the appropriate locations.
- Either 'END DATA' or 'OLD DATA' in column 1-8 on the card after the last card of actual data, will cause UNAMIT/UGETIT to stop reading and return.
- Up to 30 changeable variables are allowed.

INDEC/ININT/INALF/INNEW. TRAID has a format-free input routine which allows the mixing of descriptive text with numerical data, and also allows repetition factors.

You call INDEC or ININT with an ARGument location and a quantity N: cards will be read and interpreted until N data-values have been accumulated and stored in ARG (they will have been converted to integers if you called ININT). Only legal number fields are considered; a number field will be ended on a blank or any illegal character, such as a letter ≠ E, a second dot, etc. If a number field ends on a left parenthesis, that number is interpreted as a repetition factor, which applies to everything between that left parenthesis and the next right parenthesis that terminates a number field. On a subsequent call, INDEC/ININT resumes scanning the card-image where the scan was left off during the previous call, and reads new cards as needed to accumulate N values.

Entry INALF returns N words of BCD data from input cards. For each field, leading blanks are skipped, and the field ends on (1) the 10th character, (2) a comma, (3) a blank, or (4) column 80 of the data-card.

Entry INNEW sets a switch that prohibits the resumption of scanning of the current card -- so that on the next call INDEC, ININT, or INALF will read a new card instead of proceeding with the current one.

TITLIN. This routine reads a title card in 8A10 format; the title might then be used in calls to TITLER or SUBHEAD.

INCOL. Subroutine INCOL reads data by a format which corresponds to the SETUP array assembled by subroutine OUTSET. Data may be converted by F, I, A, or O-conversion (see OUTSET). Note that INCOL assumes DIMENSION DATA (N, LINES) where N is implied by the contents of SETUP. If INCOL finds an END DATA card, it stops reading and resets LINES to the actual number of lines read.

CHEKFIL. Function CHEKFIL checks for an endfile and (1) if there isn't any, returns, or (2) if there is one, then exits to the monitor or, if control flag IFEOF is set, returns. Typical usage is:

```
READ (5,f) list
IF (CHEKFIL (5) ) return control
```

OLDATA. This is the subroutine which TRAID input routines call, upon finding 'OLD DATA' in column 1-8 on a data-card.

3.3 OUTPUT

Principles of operation:

- TRAID output routines count lines-printed and skip automatically to a new page when appropriate. The number of lines per page may be specified on the data-card read during the special call of HEAD (see Sec. 5.7); if not specified, HEAD sets it to 57 lines per page.
- Some output routines accept a BCD title, to be printed, usually above a table of data. The first word of this title:
  - If blank, causes no printing but a 2-line skip.
  - If 1H*, in some cases, prevents all the printing which would normally be done upon this CALL.
  - If an ordinary string of characters, implies that this is a bona fide title which either is 14 words long or is terminated by a dash (after the 10th character).
- Some output routines use the SETUP array which is assembled by subroutine OUTSET. The SETUP array contains an output-format and BCD column-headings. It is the mechanism by which TRAID produces its standardized tabular printout, which is diagrammed on the next page. By selective calls to OUTSET, any or all of the 10 data-fields may be used.
- Some routines allow suppression of all printing (see TITLE = 1H*, above) or suppression of parts of the printout, as follows:
  - TITLE = blank, described above.
  - LINES (the number of vectors to be printed), if negative, suppresses printing of TITLE and column-headings.
  - LINES = 0 suppresses printing of the data, but TITLE and column-headings will be printed.
  - All output is written on logical unit 6, except by SETPLOT which the calling program controls.

DIAGRAM OF TYPICAL TRAID
TABULAR-PRINTOUT FORMAT

←120 print-columns→

BCD TITLE ENDING ON A DASH -

HEADING FOR FIELD 1	HEADING FOR FIELD 2	HEADING FOR FIELD 3	etc	etc	HEADING FOR FIELD 10
xxx	x.xx	xx			xxxxx
xxx	x.xx	xx			xxxxx
←12 col→	←12 col→	←12 col→			←12 col→

NUMERIC CONVERSION IN EACH DATA-FIELD MAY BE:

E12.4	b±9.1234±123
E11.2,1X	bb±9.12±123b
F11.6,1X	±999.123456b
F11.3,1X	±999999.123b
F10.0,2X	±999999999bb
I8,4X	±9999999bbbb
I10,2X	±999999999bb
012	777777777777
2XA8,2X	bbABCDEFGHbb
1XA8,A2,1X	bABCDEFGHPQb
(see subroutine OUTSET)	

- Some routines allow vectors to be identified by 30-character names, from an array NAMES (5, LINES).
- Routines are provided which you may call in order to skip lines or pages.
- It is intended that in ordinary circumstances, your program need not contain its own WRITE statements and formats, but should let TRAID do the printing.

ORIO. Subroutine ORIO prints (or punches) orbital-element sets. When printing, the vectors may be identified by (1) time in ORBEL(1), (2) BCD name in ORBEL(1), or (3) the vector's sequence-number. When punching (KR = 2), ORIO produces cards which can be read-in when ORIO is called by a later program (with KR = 1).

STOUT, STALE. Subroutine STOUT is designed to print a set of state vectors, either in standard form or a few variations thereof. The vectors may be identified by 30-character names, and further identified by (1) time in STATE(1), (2) BCD name in STATE(1), or (3) the vector's sequence-number. Standard form state vectors may be printed in x-y-z coordinates (KP = 1). Printing of nonstandard forms is described in the following table:

Code	Coords for Printout	Form of STATE	How STATE is transformed from the standard form
KP=2	r,θ,ϕ	polar	CALL STREP (STNDRD,STATE,+N,X)
KP=3	R,A,E	polar	" " " " " "
KP=4	alt,long,lat	polar	" " " " " "
KP,KV, & KA>5	(as for KX-5)	(special)	User's program places a position-deviation in STATE(5-7) or a velocity-deviation in STATE(8-10).
KV=5	r,θ,ϕ	polar	CALL STREP (STNDRD,STATE,+N,X)

STOUT performs half the work of printing; it calls STALE to calculate the scaled-for-output vectors.

LSKIP. This is the line-skipping routine.

HEAD. When you CALL HEAD(N), this routine (1) checks against the time and page limits and exits if appropriate, (2) skips to a new page and writes a page-heading, and (3) skips N lines on the new page. If N is zero, HEAD prints a run-end page, and returns.

SUBHEAD. Subroutine SUBHEAD is called by HEAD just after printing the heading on a new page--and SUBHEAD prints any subtitles that have been established. The way to request a subtitle is to CALL SUBHEAD (n, BCDSUBTITLE) where n is the subtitle number ($\leq$ 3). Subtitles may be changed or deleted by other calls of SUBHEAD.

CEASE. Called with a BCD message, CEASE will write the message and terminate your program's execution.

TITLER. This routine will print a message; you specify how many columns it is to be indented, and you may either allow (+) or suppress (-) centering in the remaining columns.

COUNOUT. There are two special situations in which COUNOUT is useful.

Before your program prints N lines of output (on logical unit 6), calling COUNOUT(N) will cause the line-count to be updated, and the old page to be ejected and a new one titled if necessary.

If your program is designed to input-compute-print for an unknown number of successive cases, and you want the next case's input to appear on a new page, then you should CALL COUNOUT(-60) before recycling to the input routine. To illustrate:

```
      input common data
   10 input data for a case
      compute and print for this case
      CALL  COUNOUT(-60)
      GO TO 10
```

RITEF/RITEI/RITEA/RITEO. Designed for debugging-output purposes, this routine prints a 10-character description plus a specified number of data-values, converted by E-or-F, I, A, or 0 conversion respectively. This routine may be turned off or turned back on by calling RITEOFF or RITEON; and it counts and prints (1) the total number of times called and (2) the number of times called with the current 10-character name.

OUT1, OUTN. To write a short message (ending on the 40th character or a dash) and either 1 or N data-values, you may call OUT1 or OUTN respectively. The output-conversion formats are chosen by the same code as used in subroutine OUTSET. In subroutine OUTN, sequential data-values are normally printed left-to-right, five values per line; if N is negative, however, they will be arranged top-to-bottom in five columns.

RYDIT, WRITIT. These two routines can be used to insert and print a few data-values anywhere in a line of descriptive text. This text is a string of any BCD characters; the text-string should contain substrings of zeros where you want successive data-values to be printed. [These substrings may contain a decimal, but they must begin with a zero.] Both RYDIT and WRITIT proceed to encode a data-value into BCD characters, inserting them into the text-string and taking proper account of the size of the data-value versus the size of the zero-string and the location of the decimal if any. One data-value is thus inserted into the text-string each time the routine is called; the text-plus-all-values is printed when it is completed. Differences in usage are:

- RYDIT requires that the mode (real or integer) of the data-value be specified; WRITIT deduces the mode from the apparent magnitude.
- WRITIT's interpretation of INDENT is the same as in TITLER; RYDIT does not center, but does simple indenting only.
- The text-string for RYDIT may be up to 120 characters long and it must end on an entire blank word; WRITIT allows 136 characters in length and termination on either a blank word or a dash.
- RYDIT must be instructed "do not print, but wait until the text-string is filled" (by setting MODE negative); WRITIT searches the text for unfilled zero-strings and waits-or-prints automatically.
- RYDIT inserts the encoded data-values directly into the text-string, which is an array stored in the calling program -- this means that (1) RYDIT may be used to assemble more than one text-string concurrently, but (2) after a given text has been filled with data and printed, its zero-strings must be restored for subsequent use. WRITIT, on the other hand, saves the text-string internally; this means that it must not be used "in parallel," but must be allowed to completely fill and print one text before starting on the next one (or restarting on the same one).

WHEN. This routine will help if you are investigating the computer-time usage of a program. For both central processor and peripheral processors, it prints both the cumulative time (since beginning of job) and the elapsed time (since the last call). It also prints the sequential number of this CALL, the name of, and the relative address in the calling routine.

OUTCOL. Subroutine OUTCOL is the routine which prints arrays by the format, and with headings, in a SETUP array. OUTCOL assumes that the data is in an array dimensioned (N,LINES) where N is implied by the format in SETUP.

OUTSET. This is the subroutine which assembles the 82-word SETUP array, containing an output format followed by printfield-headings for up to ten 12-column fields.

3.4 VECTORS AND COORDINATES

The TRAID routines which manipulate vectors are here divided into (a) those which work on 3-component vectors only, and (b) those which work on standard and nonstandard-form state vectors.

1. 3-Vectors Only

DOT. Dot-product.

CROSS. Cross-product.

CROS1. Cross-product, made unit-length.

UNITV. To make unit-length.

XMAG. Length of vector.

AXVEC. Sets up either Kth coordinate vector, or (if K = 0) null vector.

PROJ. Projection of a vector onto a given direction.

SEPA. Angle between vectors.

SUBVEC. Difference between vectors.

VECLIN. Linear combination of two vectors, stored in a third.

VECSUM. Linear combination of two vectors, added to a third.

AZF. Azimuth angle between vector and the x-axis.

ELF. Elevation angle between vector and the xy-plane.

LOCLAX. Sets up a triplet of unit vectors, aligned with given vectors -- e.g., local coordinates at radar on spherical earth.

2. 10-Vectors

ALTF. Altitude of a state-vector-position.

GRAV. Returns gravitational acceleration (3-vector) for a given state-position.

RADAR. Produces a set of noisy radar observations, using as inputs: (1) a set of object positions, (2) a radar location, and (3) a radar-error vector. The error-vector is interpreted in radar polar coordinates. The radar observations are returned in either (1) absolute x-y-z coordinates, or radar polar coordinates, based on the radar, with (2) azimuth measured from the x-axis and elevation measured from the xy-plane, or (3) azimuth measured from east toward north and elevation measured from the horizon.

STREP. Transforms a set of vectors between standard state-vector form and polar-coordinate form (t, r, θ, ϕ, $\dot{r}$, $\dot{\theta}$, $\dot{\phi}$, $\ddot{r}$, $\ddot{\theta}$, $\ddot{\phi}$).

SITEP. Performs earth-rotation on a state vector.

EULANG. Produces direction-cosines (10 by 3) from Euler angles, and vice versa.

TRNSFM. Transforms a state vector, using (10 by 3) direction-cosine array.

3.5 TRAJECTORY INTEGRATION

Principles of Operation:

- TRAID trajectory integration takes an initial state-vector and produces subsequent state vectors which are stored adjacent to the first.
- The user specifies the guidance commands -- these may be preselected or computed dynamically.
- The user specifies the vehicle's characteristics, such as thrust, mass, lift, drag, response to commands, etc.
- The user specifies a terminating condition for the flight -- e.g., "stop when ALTITUDE is 40 km," or "stop when TIME INTerval is 3.0 seconds."
- The flight may be computed by rectangular (KSTINT = 0) or Runge-Kutta (KSTINT = 1) integration -- KSTINT may be input on HEAD's special data-card, or set by calling SETINTG.
- Flight may go forward or backward (DTS negative in call of FLIGHT) in time. (Each state vector has the appropriate time stored in its first component, of course.)

FLIGHT. This is the executive routine for the trajectory integration task. FLIGHT must be called with three subprogram names as arguments:

- GUIDE, a subroutine which accepts as inputs CP and the current state-vector, and returns a guidance command in CP(1-3), a time-step, and an iteration counter. For this purpose you may use FLIER (described below) or you may write your own routine.
- AXEL, a subroutine which accepts a guidance command in CP(1-3), and stores an actual acceleration in components 8-10 of the current state vector. Typically, AXEL accounts for thrust, lift, drag, and response. For this purpose you may use either FLAC or RV (described below) or you may write your own routine.
- GOALFN, a function which returns the numerical value, from the current state vector, of the parameter defined by NGOAL. For this purpose you may use FOAL (see below), or you may write your own function.

When passing subprogram names as arguments (FLIER, FOAL, etc.) you must not forget to declare them in an EXTERNAL statement.

In the call of FLIGHT, your program supplies a STATES array, (1) of which the first state vector is the beginning state of the trajectory integration, and (2) into which FLIGHT inserts (some of) the successive state vectors as the integration proceeds. The number of vectors inserted into STATES is controlled by DTS, NOUT, and of course the nature of the vehicle, the trajectory, and the terminating goal.

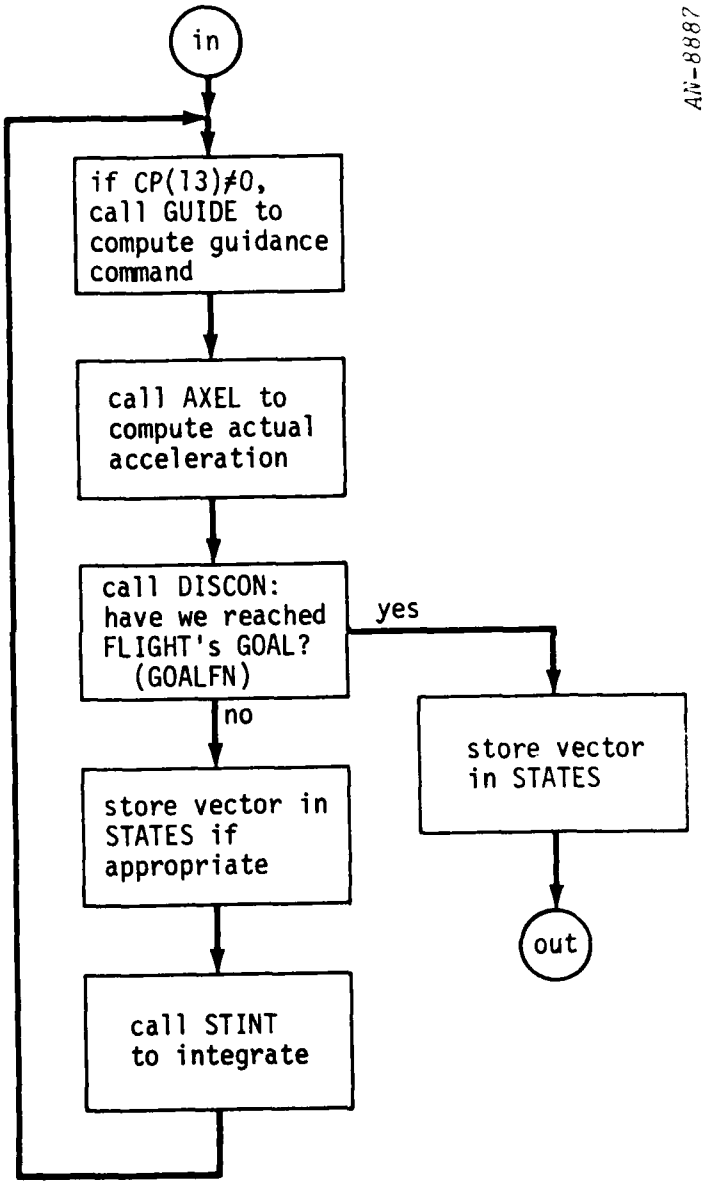
Since FLIGHT integrates until your goal is reached, there exists the danger of over-filling the STATES array. To help diagnose this condition, it is recommended that before calling FLIGHT, you set the last argument NSTATE = the number of vectors that will fit in the STATES array. Then if the number of vectors inserted into STATES exceeds that initial value of NSTATE, FLIGHT prints a message -- notice that FLIGHT keeps on integrating until the trajectory-goal is reached (or until something gets clobbered by the overflowing STATES array).

While integrating, there will be discontinuities in the trajectory -- one upon reaching your terminating goal, and perhaps others due to sudden changes in the guidance command. On these occasions, FLIGHT integrates past the discontinuity and then tries to fly back to it. It is necessary therefore that the GOAL-criteria have a continuous first derivative in the region of the discontinuity. To illustrate, a GOAL of velocity elevation = 90° will never work (because no heading has a velocity elevation of $90 + \varepsilon$), and a GOAL of velocity elevation = 89° will work reliably only if the product of turning rate (deg/s) and time-step (s) is less than 1°.

If NOUT is positive, the state vector at the above discontinuities, plus the states at stage separation and ignition times, will be stored in the STATES array (as well as the periodic states every NOUT time steps). For NOUT negative, all of these extraordinary states are omitted: and if NOUT is zero, only two vectors are returned: the initial and final states.

A simplified flow-chart of FLIGHT's operation is on the next page.

SIMPLIFIED FLOW CHART FOR FLIGHT



AW-8887

FLIER. This routine is designed to perform the GUIDE function for FLIGHT. It interprets the preselected maneuver commands (which have been read into the CP array by FLIN, called with third-digit MP = 1) in trajectory coordinates, and sets up in CP(1-3) the commanded acceleration in x-y-z coordinates. The CP array must be at least 13 words long, even if no maneuvers are specified -- if there are N maneuvers, the length of CP must be 20 + 5•N. FLIER allows sudden changes in the guidance command; therefore it uses an iteration-control flag to communicate with FLIGHT, and it returns a time-step DTE designed to reach the guidance-discontinuity. Note that FLIER uses FOAL. If FLIER finishes a maneuver-phase and finds that the next phase is undefined, it sets CP(13) to zero so that FLIGHT will not call FLIER anymore.

FLAC, RV. These routines are both intended as AXEL-routines in FLIGHT -- both set up an actual-acceleration which includes commanded acceleration and vehicle performance characteristics. Subroutine RV computes this actual acceleration for bodies with no thrust and no response, but only drag; when using RV, the C-variable does not need to be an array but may be a single number--namely the ballistic parameter in kg/m^2 if KUNITS=0, or in lbf/ft^2 if KUNITS=1. (Hint--you can input this (in lbf/ft^2) via IN1 if you punch 'PSF' in column 51-53 on the data-card.)

Subroutine FLAC provides a much broader vehicle-model than RV does. It uses the C and CP arrays to determine thrust, mass, maneuver limits, and response, and calls DNSITY, GRAV and SONIC for models of atmospheric density, gravity vector, and sound speed. For aerodynamic coefficients, FLAC uses functions CAXIAL and CNORML (which in turn refer to the C-array to compute the axial and normal-force coefficients). CAXIAL and CNORML are provided in the TRAID library, but of course you may write your own. Note that if you do write your own, and if you want to use a larger C-array, you should:

- Remember to dimension C appropriately in your main program
- Not tamper with any of the existing C-parameters 1-6, 11-12, 15-16.

- Input any extra parameters by calling FLIN with second-digit LD = 5 (don't forget the IN1-type card which specifies how many extra parameters follow).

FOAL. This function is designed to perform the GOALFN task in FLIGHT. It interprets the termination criterion NGOAL, then computes the current numerical value of the corresponding parameter, from the state vector. Note that FOAL is used explicitly by FLIER, while FLIGHT uses whatever function is specified in the CALL statement -- which usually is FOAL again, but may be another function if you want to write one. Most of the values FOAL returns are continuous functions; two kinds are not continuous:

Value is between →	and →	NGOAL criterion
$-\pi/2$	$+\pi/2$	PHI ELEVATIO LATITUDE VEL. EL. ACC. EL.
$-\pi$	$+\pi$	THETA AZIMUTH LONGITUD VEL. AZ. ACC. AZ.

CAXIAL, CNORML. These functions produce axial and normal-force coefficients from mach-number, attack-angle, the C-array, and stage number. [See note under FLAC (above) about writing your own versions of these.]

DISCON. This routine chooses a time-step to try to reach a trajectory discontinuity.

STINT. This is the routine that performs the integration by either rectangular or Runge-Kutta methods (controlled by KSTINT which is in common-block BASCON and is read from HEAD's special data-card). STINT does all its communication with FLIGHT through COMMON variables, not through arguments.

DNSITY, SONIC, GRAV. These routines provide FLIGHT with information about the real world. Note that DNSITY and GRAV are sensitive to KOORD.

UPSTATE. This routine advances a state vector to a given time, using constant acceleration.

TRPSTA. This subroutine interpolates between a pair of state vectors, to establish another state vector at a given time.

3.6 ORBITS

TRAID computes Keplerian orbits using an orbital-element 10-vector; the first of these is not used by TRAID except it may be input and output, the tenth is not used either, and the second through ninth form a redundant set describing the orbital ellipse or hyperbola. Routines which establish orbital-element vectors (ORB1, ORB2, RORB2, and ORIO) always set up the whole redundant set, so that you may use any convenient combination of them.

ORB1, ORB2, RORB2. These routines calculate orbital elements from state vectors: ORB1 uses STATE (1-7); ORB2 uses STATE1 (1-4), STATE2 (2-4), and one other parameter (for which you have four options); RORB2 uses the same inputs as ORB2, but you have six options. Both ORB2 and RORB2 store the appropriate time in STATE2 (1).

ORBP, ORBTIME, ORBTIM, ORBPD. These routines use orbital elements as inputs and produce various answers in state-vector space. ORBP, with one other parameter (your choice of six), produces a STATE(1-7); ORBTIME, with another parameter (your choice of six) produces clock time; ORBTIM, with another parameter (choice of four), produces time-since-perigee; ORBPD, with time, produces partial derivatives (dimensioned 6×6) of STATE (2-7) with respect to ORBEL (2-7).

3.7 MISCELLANEOUS

DATEF. Returns BCD date in the form 'MM/DD/YY'.

DTIMEF. Returns BCD time-of-day in the form 'HH·MM·SS'.

KALLER. Returns the name of, and relative address in, the N·th level calling routine.

LOC2. Returns (address of 2nd arg) - (address of 1st arg) + 1. Typical usage:

```
N = LOC2 (START, END)
CALL XMIT (-N, 0, START)
```

LOCFL. Returns (field length) - (address of arg) + 1.

MIXER, XMIXER. Provide a convenient way to store integers in floating-point arrays, and vice versa.

FDIV. Replaces an illegal operand (infinite or indefinite) with a zero.

PACBIT. Packs and unpacks data.

KSHFT, SHFT. Shifts a word to right or left, filling with zeros.

INDMAX, INDMIN, INDEQU. May be used to scan a list, returning the index of:

- INDMAX--the numerically largest element (the first one if there are more than one of the largest value);
- INDMIN--the numerically smallest element (the first one if there are more than one of the smallest value);
- INDEQU--the first element that matches a given match word.

RNV. Returns a quasi-normally distributed variable.

KONVERG. This routine does the tedious part of an iteration, choosing new values for the independent variable, and testing for divergence and excessive tries. KONVERG is initialized by a call of KONVSET; KONVERG stores its information in the calling program and therefore you can stack any number of iteration loops.

MINMIZE. This function works just like KONVERG except that it searches for a minimum value instead of a zero value, and that it assumes the function is parabolic (near the minimum) instead of linear (near zero).

TFLITE, TFLYTE. Returns a flight-time between two points: TFLITE assumes that (1) thrust and drag may be adequately represented by an initial speed and time-to-attain-that-speed, and (2) gravity is constant; TLFYTE dispenses with the second assumption, but is only one-quarter as fast in execution.

SONIC, DNSITY. Provide models for sound-speed and atmospheric density.

TRPLATE. Performs N-point interpolation in a monotonic list.

SETSCAL. Computes appropriate endpoints for the axes of a plot.

SETPLOT, PLOTT. Produce plots on the printer. The primary differences are:

- SETPLOT must be called with each x-y pair (data need not be stored in a table) and a character to be plotted (63 choices including blank). Characters may be added below or to the right of the previous character, which allows building up labels. Any number of traces may be plotted. SETPLOT uses a COMMON block of 663 cells.
- PLOTT works from a table of data, and assigns to each trace (up to 6 traces) a preset character. Once your data is arranged into tabular form, you can easily plot various columns versus each other, and you can easily plot a histogram instead of a single trace.

MATDIAG. Sets up a square matrix with a given vector on the diagonal.

MATSCAL. Multiplies a matrix by a scalar.

MATFLIP. Transposes a square matrix (into the same location).

MATRANS. Transposes a matrix, storing in another matrix.

MATNVRT. Inverts a matrix

MATADD, MATSUB. Add and subtract matrices.

MATMULT. Multiplies two matrices.

MATIGEN. Calculates eigenvalues and eigenvectors.

MATPANP. Produces a submatrix by selecting and reordering specified rows and columns from the original matrix.

JACOBI. Is a more general form of MATIGEN.

4.0 TRAID SPECIFICS

In this section are the complete FORTRAN listings of all TRAID routines. Information which was too subtle for the Primer (Sec. 2.0) or was too detailed for the Generalities (Sec. 3.0), will be found in this section. With luck, specific questions may be answered by the explanatory comments in each routine; as a last resort the answers can be found in the source code itself.

```
      FUNCTION ALTF(STATE)
C     SOURCE DATE 67.0726   BRAND NEW CODE
C
C     RETURNS ALTITUDE OF POSITION IN STATE(2-4)
C
CCOMPKG.BASCON
000003      COMMON /BASCON/
           1   PROGRM,    KPAGE,    LINE,     TOFDAY,   RUN,      RUNIN(6),
           2   MSG,       FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
           3   KLASS,     KGROUP,   KUNITS,   KSTINT,   KOORD,    IFEOF
           4  ,NDFILE
C
CCOMPKG.CONCON
000003      COMMON /CONCON/
           1   PI,        SRD,      SLV,      SMF,      SKP,      RBODY,
           2   GACC,      GCON,     MBODY,    MNOZRO,   TWOPI,    MAFPI
C
000003      DIMENSION STATE(10),P(3)
C
000003      CALL AMIT(3,STATE(2),P)
000006      IF(KOORD-1) 1,2,3
000011    1 A = P(3)
000013      GO TO 4
000013    2 P(3) = P(3)+RBODY
000015    3 A = XMAG(P)-RBODY
000021    4 ALTF = A
000023      RETURN
000023      END  ALTF
```

PRECEDING PAGE BLANK-NOT FILMED

```
                      SUBROUTINE  AXVEC (K,V)
                C     SOURCE DATE 67.0720    BRAND NEW CODE
                C
                C     RETURNS VECTOR  V = UNIT VECTOR IN  K-DIRECTION
                C                     OR  V = NULL-VECTOR IF K = 0
                C
                C
      000004          DIMENSION V(3)
                C
      000004          V(1)=V(2)=V(3)=0.
      000007          IF(K.GT.0) V(K)=1.
      000012          RETURN
      000013          END  AXVEC
```

```
      FUNCTION AZF(X)                                                    10
      SOURCE DATE 66.0101   BRAND NEW CODE                               11
C                                                                        12
C     RETURNS THE AZIMUTH ANGLE OF VECTOR X
C
C     ANGLE MEASURED FROM X-AXIS, POSITIVE TOWARD Y-AXIS                 13
C     ITS VALUE IS BETWEEN -PI AND +PI.
C                                                                        14
C     DIMENSION X(3)                                                     15
C
000003      AZF=0.
000003      IF(X(1).EQ.0..AND.X(2).EQ.0.) RETURN
000003      AZF=ATAN2(X(2),X(1))
000013      RETURN
000020      END  AZF
000020
```

```
                      FUNCTION CAXIAL (VMACH,ALF,C,K,CP)                                    68.0311
                C     SOURCE DATE 68.1205   USE FDIV WHEN DENOMINATOR MAY BE ZERO
                C     SOURCE DATE 68.0311   ADD 5-TH ARGUMENT
                C     SOURCE DATE 68.0310   CHANGE DRAG PROFILE TO FORM  CX=Z+X/(VMACH-Y)
                C     SOURCE DATE 67.0707   ADAPTED FROM FLAC OF 66.0601
                C
                C     RETURNS AXIAL -FORCE COEFFICIENT, AT MACH NUMBER VMACH AND
                C     ATTACK ANGLE ALF, FOR A VEHICLE DESCRIBED BY THE K-TH STAGE
                C     OF THE C ARRAY
                C
000010                DIMENSION C(3,16)
000010                DATA (V1=0.5), (V2=1.0), (V3=3.), (V4=10.)
                C
000010                IF(VMACH.GT.V1) GO TO 2
000013                CAXIAL=C(K,7)  $  RETURN
000015              2 IF(VMACH.GT.V2) GO TO 4
000021                CAXIAL=C(K,7)+(VMACH-V1)/(V2-V1)*(C(K,8)-C(K,7))  $  RETURN
000032              4 Z=(V4-V3)/(V3-V2)*FDIV((C(K,8)-C(K,9))/(C(K,9)-C(K,10)))          68.1205
000054                Y=FDIV((Z-V4)/(Z-V2))                                             68.1205
000063                X=(C(K,8)-C(K,9))*(V3-Y)*(V2-Y)/(V3-V2)
000100                Z=C(K,8)-FDIV(X/(V2-Y))                                           68.1205
000113                CAXIAL=Z+FDIV(X/(VMACH-Y))                                        68.1205
000124                RETURN
000125                END  CAXIAL
```

```
      SUBROUTINE CEASE (MESSAGE)
      SOURCE DATE 67.0726   BRAND NEW CODE
C
C        PROVIDES A MESSAGE (AS IN TITLER) AND AN EXIT
C        ALL YOU HAVE TO DO IS CALL IT
C
C
      CALL TITLER (1,MESSAGE)
000003 CALL HEAD(0)
000006 CALL EXIT
000010 END  CEASE
000011
```

```
                FUNCTION CHEKFIL  (LUN)                                          6A.0209
      *         SOURCE DATE 68.0209       CONVERT TO CDC 6400
      C         SOURCE DATE 66.1215     NDFILE+1, AND IFEOF=0 IN DATA STMT
      C
      C         CALLS EXIT OR SETS EOF FLAG IF ENDFILE ON LOGICAL UNIT LUN
      C
      C             *** NOTE - IF AN ENDFILE HAS BEEN FOUND, THEN CHEKFIL WILL
      C                        EXIT UNLESS -IFEOF- HAS BEEN SET NON-ZERO
      C
      C             TO LEARN NUMBER OF ENDFILES WHICH HAVE BEEN READ,
      C                  N=NUMFILS(DUMMY)
      C             TO RESET TO L THE NUMBER OF ENDFILES,
      C                  CALL SETNFIL(L)
      C             TO RESET TO L THE ENDFILE-CONTROL SWITCH IFEOF,
      C                  CALL SETIFEF(L)
      C
      CCOMPKG,BASCON
000003                COMMON /BASCON/
                     1   PROGAM,    KPAGE,    LINE,     TOFDAY,   RUN,      RUNID(6),
                     2   MSG,      FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
                     3   KLASS,    KGROUP,   KUNITS,   KSTINT,   KOORD,    IFEOF
                     4  ,NDFILE
      C
000003                DATA (NDFILE=0),(IFEOF=0)                                  66.1215
000003                EQUIVALENCE (AEQ,IEQ)
      C
000003                CHEKFIL =0.                                                68.0209
000004                IF(EOF,LUN) 10,40
000007              10 IF(IFEOF) 30,20
000012              20 CALL MEAD(0)
000014                 CALL EXIT
000015              30 NDFILE=NDFILE+1                                           66.1215
000017                 CHEKFIL =1.0                                              68.0209
000020              40 RETURN
      C
000022                ENTRY NUMFIL S                                             68.0209
000030                IEQ=NDFILE
000032                CHEKFIL =AEQ                                               68.0209
000033                RETURN
      C
000034                ENTRY SETIFE F                                             6A.0209
000042                IFEOF=LUN
000044                RETURN
      C
000046                ENTRY SETNFIL                                              6A.0209
000054                NDFILE=LUN
000055                RETURN
      C
000057                END  CHEKFIL
```

PRECEDING PAGE BLANK-NOT FILMED

```
                FUNCTION CNORML (VMACH,ALF,C,K,CP)                              68.0311
          C     SOURCE DATE 68.0311    ADD 5-TH ARGUMENT
          C     SOURCE DATE 67.0707    ADAPTED FROM FLAC OF 66.0601
          C
          C     RETURNS NORMAL-FORCE COEFFICIENT, AT MACH NUMBER VMACH AND
          C     ATTACK ANGLE ALF, FOR A VEHICLE DESCRIBED BY THE K-TH STAGE
          C     OF THE C ARRAY
          C
000006            DIMENSION C(3,16)
          C
000006            CNORML=ALF*(C(K,13)+ALF*C(K,14))
000013            RETURN
000013            END  CNORML
```

```
      PROGRAM COMPKG
C     SOURCE DATE 69.0709    WRITE ERROR MESSAGE AMID MERGED CARD IMAGES
C     SOURCE DATE 68.1205    PRINT CCOMPKG CARD BOTH BEFORE AND AFTER
C     SOURCE DATE 68.0301    HRAND NEW CODE
     X (
C          ASSIGN BUFFERS
     X     INPUT=4000,
     X     OUTPUT=2022,
     X     MERGED=4000,
     X     TAPE41=OUTPUT,
     X     TAPE1=INPUT,
     X     TAPE2=INPUT,
     X     TAPE10=MERGED)
C
*          LOADED VIA CONTROL CARD, THIS PROGRAM READS FORTRAN MACROS
*          (EG - PACKAGES OF COMMON STATEMENTS) FROM =INPUT=, THEN READS
*          FORTRAN SOURCE CODE FROM =INPUT=, AND WRITES THAT SOURCE CODE
*          (INCLUDING THE APPROPRIATE MACROS) ON FILE =MERGED=.
*          EACH MACRO IS A SET OF CARDS PRECEDED BY A NAME-CARD WITH
*          =MACRONAME= IN COL 1-10, AND FOLLOWED BY A CARD WITH =ENDCOM=
*          IN COL 1-6. THE LAST SUCH ENDCOM CARD IS FOLLOWED BY A CARD
*          WITH =ENDALL= IN COL 1-6, AND NEXT IS THE FORTRAN CODE IN WHICH
*          THE MACROS ARE TO BE INSERTED. IN PROCESSING THIS CODE,
*          THE PROGRAM LOOKS FOR CARDS WITH =CCOMPKG,= IN COL 1-8, TAKES
*          THE =MACRONAME= FROM COL 9-18, AND WRITES THE CARD IMAGES OF THE
*          CORRESPONDING MACRO, EMBEDDED AMONG THE FORTRAN CARDS, ON =MERGED=.
*          UPON READING AN ENDOFRECORD, THE PROGRAM REWINDS =MERGED= AND EXITS.
*          TYPICAL USE OF THIS PROGRAM IS IMPLIED BY THE TYPICAL DECK SETUP ...
*           CONTROL CARDS       JOB, ...
*           (REC 0)             COMPKG.
*                               RUN,S,,,MERGED.
*                               LGO.
*                               7/8/9
*           MACROS PLUS         COMNAME1
*           FORTRAN                   COMMON /A/ X,Y,Z
*           (REC 1)             ENDCOM
*                               COMNAME2
*                                     COMMON /1/ P,Q,G,R
*                                     INTEGER R  $  EQUIVALENCE (G,R)
*                               ENDCOM
*                               ENDALL
*                                     SUBROUTINE R(I,J)
*                               CCOMPKG,COMNAME2
*                               CCOMPKG,COMNAME1
*                                     DO 20 L=1,10
*                                     ETC...
*                               7/8/9
*          DATA (REC 3)         DATA CARDS
*
C
000003       DIMENSION AMRAY( 4000),LOW(62),LMI(62),NAME(62),KARD( 9)
000003       COMMON AMRAY
000003       INTEGER AMRAY
000003       DATA (MAXARAY = 4000)
000003       DIMENSION MESG(4,7)
000003       DATA(MESG =                                                    69.0709
            1             31H PARITY-CHECK ON LIM INPUT      ,            69.0709
            2             31H COMMON STATEMENTS EXCEED ARRAY,             69.0709
            3             31H PARITY-CHECK ON SOURCE INPUT   ,            69.0709
            4             31H NEVER HEARD OF      (NAME)     ,            69.0709
            5             31H COMPKG   (NAME)     INSERTED   ,            69.0709
            6             31H PARITY-CHECK ON COMMON INPUT   ,            69.0709
            7             31H TOO MANY COMMON PACKAGES       )            69.0709
C
C
000003       LUN=10  $  NUN=1  $  MUN=2
000006       KA=1
000007       NCOM = 0
000010       KRCD=BMI COMPKG  $  WRITE(61,1) KBCD
000017       CALL AMIT(1,DATEF(KDATE),KDATE)                               69.0709
C
```

PRECEDING PAGE BLANK-NOT FILMED

```
000024     10   NCOM=NCOM+1
000026          KGO=1
000027          IF(NCOM.GT.62) GO TO 814
000032          LO(NCOM)=KA+8
000034          KH=KA+7
000035      102 READ(MUN,1) (AHRAY(K),K=KA,KH)
000050        1 FORMAT(8A10)
000050          IF(EOF,MUN) 806,106
000053      106 GO TO (1062,1064,1064),KGO
000062     1062 IF(ARRAY(KA).EQ.8HENDALL  ) GO TO 116
000065     1064 KB=KA+7
000067          IF(KB.GT.MAXARRAY) GO TO 802
000072          IF(ARRAY(KA).EQ.8HENDCOM  ) GO TO 112
000074      110 KA=KH+1
000076          KB=KH+8
000077          KGO=2
000100          GO TO 102
000101      112 LHI(NCOM)=KB-H
000104          L=LO(NCOM)-8
000105          NAME(NCOM)=ARRAY(L)
000107          IF(NCOM.LT.2) GO TO 115
000111          DO 114 JC=2,NCOM
000113          IF(NAME(JC-1).EQ.NAME(NCOM)) NAME(JC-1)=-1
000117      114 CONTINUE
000122      115 IF(KGO.EQ.3) 116,10
            C
000127      116  NCOM=NCOM-1
            C
            C
000131       30 READ(MUN,2) KARD
000137        2 FORMAT(AR,7A10,A2)
000137          IF(EOF,MUN) 80,34
000142       34 WRITE(LUN,2) KARD
000150          IF(KARD.NE.8HCCOMPKG,) GO TO 30
000152          IF(NCOM.LT.,) GO TO 3H
000155          DO 36 N=1,NCOM
000156          IF(NAME(N).EQ.KARD(2)) GO TO 40
000160       36 CONTINUE
000162       3H MESG(3,4) = KARD(2)
000164          WRITE(61,1) (MESG(K,4),K=1,4)                          69.0709
000176          WRITE(LUN,1) (MESG(K,4),K=1,4)
000207          GO TO 30
            C
000210       40 KA = LO(N)
000212       44 MESG(2,5) = NAME(N)
000214          WRITE(61,1) (MESG(K,5),K=1,4)
000226       48 KB = KA+7
000230          WRITE(LUN,1) (ARRAY(K),K=KA,KB)
000243          IF(KB.GE.LHI(N)) GO TO 50                              68.1205
000246          KA = KB+1
000250          GO TO 48
            C
000250       50 WRITE(LUN,2) KARD                                      68.1205
000256          CALL XMIT(-9,1H ,KARD)                                 69.0709
000261          KARD(1)=1HC                                            69.0709
000263          KARD(H)=KDATE                                          69.0709
000264          WRITE(LUN,2) KARD                                      69.0709
000272          GO TO 30                                               68.1205
            C
            C
000273       60 END FILE LUN  $  REWIND LUN
000277          CALL EXIT
            C
            C
            C
000300      802 WRITE(61,1) (MESG(K,2),K=1,4)
000312          GO TO 810
            C
000313      804 WRITE(61,1) (MESG(K,3),K=1,4)
000325          GO TO 810
            C
000326      806 KB=KA+7
000330          KGO=3
000331          GO TO 112
            C
000332      808 WRITE(61,1) (MESG(K,6),K=1,4)
000344          GO TO 104
            C
000345      814 WRITE(61,1) (MESG(K,1),K=1,4)
            C
000357      810 CALL EXIT
000360          END  COMPKG
```

```
                      SUBROUTINE COUN OUT (LINES)                                    6R.0209
                     *SOURCE DATE 68.0209        CONVERT TO COC 6400
                     C SOURCE DATE 67.0601    BRAND NEW CODE
                     C
                     C SKIPS PAGE AND COUNTS LINES FOR PROGRAMS WHICH WRITE OUTPUT
                     C TYPICAL USE IS TO CALL COUNOUT (N) BEFORE WRITING N LINES OF OUTPUT
                     C IF N IS NEGATIVE PAGE SKIP IS SUPPRESSED - THIS MIGHT BE USED WHEN
                     C YOU WANT NEXT INPUT TO APPEAR ON NEW PAGE, BUT IF THERE IS
                     C NO MORE INPUT, YOU WANT TO AVOID THE EXTRA PAGE.
                     C
                     CCOMPKG,BASCON
000003                      COMMON /BASCON/
                           1   PROGNM,    KPAGE,     LINE,      TOFDAY,    RUN,       RUNID(6),
                           2   MSG,       FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
                           3   KLASS,     KGROUP,    KUNITS,    KSTINT,    KOORD,     IFEOF
                           4  ,NOFILE
                     C
                     C
000003                      IF(LINE+LINES.GT.MAXLN) CALL HEAD(2)
000011                      LINE=LINE+IABS(LINES)
000013                      RETURN
000014                      END  COUNOUT
```

```
                      SUBROUTINE CROSS (A,B,C)
                C     SOURCE DATE 67.0711   BRAND NEW CODE
                C
                C     RETURNS C, THE CROSS PRODUCT  A X B
                C     FOR ENTRY CROS1, RETURNS C UNITIZED
                C
000005                DIMENSION A(3),B(3),C(3)
                C
000005                KROSS=1
000006                GO TO 10
                C
000006                ENTRY CROS1
000015              5 KROSS=0
000016             10 C1=A(2)*B(3)-A(3)*B(2)
000024                C2=A(3)*B(1)-A(1)*B(3)
000030                C(3)=A(1)*B(2)-A(2)*B(1)
000035                C(2)=C2
000037                C(1)=C1
000040                IF(KROSS) RETURN
000042                C1=SQRT(C1**2+C2**2+C(3)**2)
000052                C(1)=C(1)/C1
000054                C(2)=C(2)/C1
000055                C(3)=C(3)/C1
000057                RETURN
000057                END  CROSS
```

```
VER 1.1       DATEF                                                    08/30/08

                                          IDENT     DATEF
                              000010      PROGRAM LENGTH

                                          BLOCKS

                        000000  000010    PROGRAM*  LOCAL

                                          ENTRY POINTS

                                             000001 DATEF

                                          EXTERNAL SYMBOLS

                                          CPC

                                          *     SOURCE DATE 68.0200       CONVERT TO CDC 6400
                                          *     RETURNS DATE IN 10 CHARACTERS
                                                ENTRY     DATEF
000000  0401240506000000000001                VFD       30/9LDATEF,30/1
000001                                  DATEF BSSZ      1
000002  76600                                 SX6       B0
              5160000007 *                    SA6       =SANS
000003  0100000000 X                           RJ        =XCPC
000004  241115000001000000007 *                VFD       18/3CTIM,1/1,1/1,4/0,12/1,6/0,18/ANS
000005  5150000007 *                           SA5       =SANS
                    0305000005 *                ZR        X5,*
000006  10655                                   BX6       X5
              0200000001 *                      JP        DATEF
000010                                          END

                              034511    UNUSED STORAGE          15 STATEMENTS        3  SYMBOLS
```

```
                SUBROUTINE DISCON(GOAL,GNOW,GLAST,KONDX,DT,DTE)                  42
C               SOURCE DATE  69.0408   BRAND NEW CODE, RE-WRITTEN TO OLD SPECS
C               SOURCE DATE 69.032R   SIMPLIFY GOAL/GLAST/GNOW TEST
*               SOURCE DATE 68.0209       CONVERT TO CDC 6400
C               SOURCE DATE 64.0824
C                                                                                43
C                   ADJUSTS TIME STEP TO ITERATE BACK TOWARD A DISCONTINUITY
C                                                                                45
C                   GOAL    -  NUMERICAL VALUE OF CRITERION AT WHICH INTEGRATION IS   46
C                              TO BE INTERRUPTED                                  47
C                   GNOW    -  PRESENT VALUE OF GOAL                           48
C                   GLAST   -  PREVIOUS VALUE OF GOAL                          49
C                   KONDX   -  CURRENT CONDITION OF ITERATION, SET BY DISCON TO  -   50
C                              POSITIVE - ITERATION IN PROGRESS, KONDX = STEP NO.   51
C                              NEGATIVE - ITERATION COMPLETED THIS TIME STEP      52
C                              ZERO - NO ITERATION IN PROGRESS (MUST BE INITIAL   53
C                              VALUE OF KONDX SUPPLIED TO DISCON)                 54
C                   DT      -  LAST ACTUAL TIME STEP                           55
C                   DTE     -  TIME STEP ESTIMATED TO REACH DISCONTINUITY       56
C                                                                                57
C                                                                                60
000011          DG=ABS(GOAL-GNOW)
000013          TLRNC=1.E-5 * AMAX1(GOAL,1.)
000017          IF(KONDX) 2,4,10
000020        2 KONDX=0
000021        4 IF(DG.LT.TLRNC) GO TO 12
000024          IF((GOAL-GNOW)*(GOAL-GLAST).GT.0.) GO TO 8
000031        6 KONDX=KONDX+1
000032          DTE=DT*FDIV((GOAL-GNOW)/(GNOW-GLAST))
000045          IF(DTE.EQ.0.) DTE=10.*DT
000047        8 GLAST=GNOW
000050          RETURN
C
000051       10 IF(DG.GT.TLRNC) GO TO 6
000055       12 KONDX=-1
000056          GO TO 8
C
000057          END  DISCON
```

PRECEDING PAGE BLANK-NOT FILM

```
                      FUNCTION DNSITY(S)                                                    77
                C     SOURCE DATE 68.0209       USE ALTF FOR ALTITUDE
                C     SOURCE DATE 64.0728
                C                                                                           78
                C         RETURNS ATMOSPHERIC DENSITY FOR STATE VECTOR S                    79
                C                                                                           80
                C         WRITTEN 7/28/64                                                   81
                C                                                                           82
                      CCOMPKG,BASCON
000003                COMMON /BASCON/
                     1   PROGRM,   KPAGE,    LINE,     TOFDAY,   RUN,      RUNID(6),
                     2   MSG,      FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
                     3   KLASS,    KGROUP,   KUNITS,   KSTINT,   KOORD,    IFEOF
                     4   ,NDFILE
                C
                      CCOMPKG,CONCON
000003                COMMON /CONCON/
                     1   PI,       SRD,      SLV,      SMF,      SKP,      RBODY,
                     2   GACC,     GCON,     WBODY,    RHOZRO,   TWOPI,    HAFPI
                C
000003                DIMENSION Z(5),A(5),B(5)                                            87
000003                DATA ((Z(I),I=1,5)=-10.,5000.,10000.,40000.,100000.)                88
000003                DATA ((A(I),I=1,5)=1.,1.,1.08004206,1.50679735,.35487133)           89
000003                DATA ((B(I),I=1,5)=0.,-.0001016,-.000117,-.0001561,-.0001185)
                C
000003                H=ALTF(S)                                                        68.0209
000006           6    IF (KUNITS.NE.0) H=H/SMF                                            96
000011                DO 7 I=1,5                                                          97
000013                IF (H.LT.Z(I)) GO TO 8                                              98
000016           7    CONTINUE                                                            99
000020                DNSITY=0.                                                         100
000021                RETURN                                                            101
000022           8    DNSITY=RHOZRO*A(I)*EXP(B(I)*H)                                    102
000033                RETURN                                                            103
000033                END  DNSITY
```

```
                      FUNCTION DOT(A,B)                                          105
                C     SOURCE DATE 66.0101   BRAND NEW CODE
                C                                                                106
                C         RETURNS THE DOT PRODUCT OF VECTOR A WITH VECTOR B       107
                C                                                                108
000004                DIMENSION A(3),B(3)                                        109
                C
000004                DOT=0.                                                     110
000005                DO 1 I=1,3                                                 111
000006           1    DOT=A(I)*B(I)+DOT                                          112
000014                RETURN                                                     113
000015                END  DOT
```

```
R 1.1       DTIMEF                                                    08/30/6R

                                        IDENT     DTIMEF
                          000010        PROGRAM LENGTH

                                        BLOCKS

                    000000    000010    PROGRAM*   LOCAL

                                        ENTRY POINTS

                                           000001 DTIMEF

                                        EXTERNAL SYMBOLS

                                        CPC

                                        *      SOURCE DATE 68.0209       CONVERT TO CDC 6400
                                        *      RETURNS TIME-OF-DAY IN 10 CHARACTERS
                                               ENTRY     DTIMEF
000000  0424111505060000000001                 VFD       36/6LDTIMEF,24/1
000001                                 DTIMEF  BSSZ      1
000002  76600                                  SX6       B0
              5160000007 *                     SA6       =SANS
000003  0100000000 X                            RJ        =XCPC
000004  2411150000020000007 *                   VFD       18/3CTIM,1/1,1/1,4/0,12/2,6/0,18/ANS
000005  5150000007 *                            SA5       =SANS
                    0305000005 *               ZR        X5,*
000006  10655                                   BX6       X5
              0200000001 *                     JP        DTIMEF
000010                                         END

                          034511    UNUSED STORAGE          15 STATEMENTS       3  SYMBOLS
```

PRECEDING PAGE BLANK-NOT FILMED

```
                                                                        134
      FUNCTION ELF(A)                                                   139
      SOURCE DATE 66.0101   BRAND NEW CODE                              140
C                                                                       141
C     RETURNS THE ELEVATION ANGLE OF VECTOR A
C
C     ANGLE MEASURED FROM X-Y PLANE, POSITIVE TOWARD Z-AXIS
C     ITS VALUE IS BETWEEN -PI AND +PI.                                 142
C
C     DIMENSION A(3),B(3)                                               143
000003                                                                  144
C
      B(2)=A(3)
000003B(1)=SQRT(A(1)*A(1)+A(2)*A(2))
000004ELF=0.                                                            145
000012IF(B(1).EQ.0..AND.B(2).EQ.0.) RETURN                              146
000013ELF=ATAN2(B(2),B(1))
000023RETURN
000027END  ELF
000027
```

```
      SUBROUTINE EULANG(MODE,AXES,ANGLES,NDERIV)                          168
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE  66.0921    REWRITTEN TO ALLOW ANGULAR RATES
C                                                                         169
C         PRODUCES EULER ANGLES FROM DIRECTION COSINES, OR VICE VERSA     150
C                                                                         151
C         MODE   -  IF POSITIVE, EULER ANGLES ARE RETURNED FROM DIREC-    152
C                   TION COSINES.  IF NEGATIVE, DIRECTION COSINES ARE     153
C                   RETURNED FROM EULER ANGLES.                           154
C         AXES   -  DIRECTION COSINE ARRAY                                155
C         ANGLES -  EULER ANGLE VECTOR                                    156
C         NDERIV -  HIGHEST ORDER OF TIME DERIVATIVE INCLUDED             157
C                                                                         158
C                                                                         160
      CCOMPKG,CONCON
000006      COMMON /CONCON/
           1   PI,        SRD,       SLV,       SMF,       SKP,       RBODY,
           2   GACC,      GCON,      WBODY,     RHO2HO,    TWOPI,     HAFPI
C
000006      DIMENSION EN(10),AXES(10,3),ANGLES(10),SNEN(4),CSEN(4)
000006      DIMENSION SNEND(4),CSEND(4),SNENDD(4),CSENDD(4),ANGLS(10)
000006      EQUIVALENCE
           1 (SNA,SNEN(2)),(SNAD,SNEND(2)),(SNADD,SNENDD(2))
           2,(SNB,SNEN(3)),(SNBD,SNEND(3)),(SNBDD,SNENDD(3))
           3,(SNG,SNEN(4)),(SNGD,SNEND(4)),(SNGDD,SNENDD(4))
           4,(CSA,CSEN(2)),(CSAD,CSEND(2)),(CSADD,CSENDD(2))
           5,(CSB,CSEN(3)),(CSBD,CSEND(3)),(CSBDD,CSENDD(3))
           6,(CSG,CSEN(4)),(CSGD,CSEND(4)),(CSGDD,CSENDD(4))
000006      EQUIVALENCE
           1    (A,ANGLS(2)),(AD,ANGLS(5)),(ADD,ANGLS(8))
           2,   (B,ANGLS(3)),(BD,ANGLS(6)),(BDD,ANGLS(9))
           3,   (G,ANGLS(4)),(GD,ANGLS(7)),(GDD,ANGLS(10))
000006      EQUIVALENCE
           1     (ENX,EN(2)),(ENXD,EN(5)),(ENXDD,EN(8))
           2,    (ENY,EN(3)),(ENYD,EN(6)),(ENYDD,EN(9))
           3,    (ENZ,EN(4)),(ENZD,EN(7)),(ENZDD,EN(10))
C
C
000006      IF(MODE.LT.0) GO TO 30
000007      ANGLES( 1 ) = AXES(1,1)
000010      CSB = AMIN1(AXES(4,3),1.)
000014      B = ACOS(CSB)
000017      SNB = SIN(B)
000021      ENX=FDIV(-AXES(3,3)/SNB)                                  68.0209
000032      ENY=FDIV(AXES(2,3)/SNB)                                   68.0209
000041      IF(CSB.LT.1.) GO TO 4
000044      ENX = AXES(2,1)
000045      ENY = AXES(3,1)
000047    4 ENZ = 0.
000050      SNA = ENX
000052      CSA = ENY
000053      A = ATAN2(CSA,SNA)
000056      G = SEPA(EN(2),AXES(2,1))
000066      IF(AXES(4,1).LT.0.) G=-G
000071      II = 4
000072      IF(NDERIV.LT.1) GO TO 28
C
000074      BD=FDIV(-AXES(7,3)/SNB)                                   68.0209
000103      IF(SNB.EQ.0.) BD=SIGN(SQRT(AXES(7,1)**2+AXES(7,2)**2),AXES(6,1))
000117      SNBD = CSB*BD
000121      ENXD = FDIV(-(AXES(6,3)-ENX*SNBD)/SNB)                    68.0209
000132      ENYD = FDIV( (AXES(5,3)-ENX*SNBD)/SNB)                    68.0209
000143      ENZD = 0.
000144      AD = FDIV(ENYD/CSA)                                       68.0209
000151      IF(CSA.EQ.0.) AD=-FDIV(ENXD/SNA)                          68.0209
000161      CSGD = DOT(EN(2),AXES(5,2)) + DOT(EN(5),AXES(2,2))
000201      GD = FDIV(-CSGD/SNG)                                      68.0209
000205      IF(SNG.EQ.0.) GD=SQRT(ENXD**2+ENYD**2)
000215      II = 7
000216      IF(NDERIV.LT.2) GO TO 28
```

PRECEDING PAGE BLANK-NOT FILMED

```
      C
000223        HDD = FDIV(-(AXES(10,3)-BD*SNBD)/SNB)                          6A.0209
000234        SNBDD = -SNB*BD**2+CSB*HDD
000241        ENXDD = FDIV(-(AXES(9,3)-2.*ENXD*SNBD-ENX*SNBDD)/SNB)          6A.0209
000254        ENYDD = FDIV(+(AXES(8,3)-2.*ENYD*SNBD-ENX*SNBDD)/SNB)          6A.0209
000270        ENZDD = 0.
000271        ADD = FDIV((ENYDD+SNA*AD**2)/CSA)                              6A.0209
000300        GDD = FDIV(-(DOT(EN(2),AXES(8,2))+2.*DOT(EN(5),AXES(5,2))
             X     + DOT(EN(8),AXES(2,2))-GD*SNGD)/SNG)                      6A.0209
000337        II = 10
      C
000340     28 DO 29 I=2,II
000342     29 ANGLES(I) = ANGLS(I)
000347        RETURN
      C
      C
000350     30 AXES(1,1) = ANGLES(1)
000354        DO 32 J=2,4
000355        SNFN(J) = SIN(ANGLES(J))
000364     32 CSFN(J) = COS(ANGLES(J))
000376        IF(NDERIV.LT.1) GO TO 40
000400        DO 34 J=2,4
000402        SNFND(J) = CSFN(J)*ANGLES(J+3)
000406     34 CSFND(J) = -SNFN(J)*ANGLES(J+3)
000414        IF(NDERIV.LT.2) GO TO 40
000416        DO 36 J=2,4
000420        SNFNDD(J) = -SNFN(J)*ANGLES(J+3)**2+CSFN(J)*ANGLES(J+6)**2
000431     36 CSFNDD(J) = -CSFN(J)*ANGLES(J+3)**2-SNFN(J)*ANGLES(J+6)**2
      C
000444     40 AXES(2,1)= CSA*CSG-SNA*CSB*SNG                                 179
000454        AXES(3,1)= SNA*CSG+CSA*CSB*SNG                                 180
000456        AXES(4,1)= SNB*SNG                                             181
000457        AXES(2,2)=-CSA*SNG-SNA*CSB*CSG                                 182
000462        AXES(3,2)=-SNA*SNG+CSA*CSB*CSG                                 183
000465        AXES(4,2)= SNB*CSG                                             184
000466        AXES(2,3)= SNA*SNB                                             185
000470        AXES(3,3)=-CSA*SNB                                             186
000471        AXES(4,3)=CSB                                                  187
000473        IF(NDERIV.LT.1) RETURN
      C
000476        PX = CSAD*SNG+CSB*SNGD
000502        PY = CSBD*CSG+CSB*CSGD
000505        AXES(5,1) = CSAD*CSG+CSA*CSGD-SNAD*CSA*SNG-SNA*PX
000515        AXES(6,1) = SNAD*CSG+SNA*CSGD-CSAD*CSB*SNG-CSA*PX
000525        AXES(7,1) = SNBD*SNG+SNB*SNGD
000530        AXES(5,2) = -CSAD*SNG-CSA*SNGD-SNAD*CSB*CSG-SNA*PY
000541        AXES(6,2) = -SNAD*SNG-SNA*SNGD- CSAD*CSB*CSG+CSA*PY
000550        AXES(7,2) = SNBD*CSG+SNB*CSGD
000554        AXES(5,3) = SNAD*SNB+SNA*SNBD
000557        AXES(6,3) = -CSAD*SNB-CSA*SNBD
000562        AXES(7,3) = CSBD
000563        IF(NDERIV.LT.2) RETURN
      C
000566        PXD = CSBDD*SNG + 2.*CSBD*SNGD + CSB*SNGDD
000575        PYD = CSBDD*CSG + 2.*CSBD*CSGD + CSB*CSGDD
000603        AXES(08,1) = CSADD*CSG+2.*CSAD*CSGD+CSA*CSGDD
             X           -SNADD*CSB*SNG-2.*SNAD*PX-SNA*PXD
000622        AXES(09,1) = SNADD*CSG+2.*SNAD*CSGD+SNA*CSGDD
             X           +CSADD*CSB*SNG+2.*CSAD*PX+CSA*PXD
000641        AXES(10,1) = SNBDD*SNG+2.*SNBD*SNGD+SNB*SNGDD
000650        AXES(08,2) = -CSADD*SNG-2.*CSAD*SNGD-CSA*SNGDD
             X           -SNADD*CSB*CSG-2.*SNAD*PY-SNA*PYD
000667        AXES(09,2) = -SNADD*SNG-2.*SNAD*SNGD-SNA*SNGDD
             X           +CSADD*CSB*CSG+2.*CSAD*PY+CSA*PYD
000706        AXES(10,2) = SNBDD*CSG+2.*SNBD*CSGD+SNB*CSGDD
000715        AXES(08,3) = SNADD*SNB+2.*SNAD*SNBD+SNA*SNBDD
000723        AXES(09,3) = -CSADD*SNB-2.*CSAD*SNBD-CSA*SNBDD
000732        AXES(10,3) = CSBDD
000734        RETURN
000734        END  EULANG
```

```
VER 1.1       FDIV                                                    08/30/68

                                        IDENT     FDIV
                            000005      PROGRAM LENGTH

                                        BLOCKS

                    000000  000005      PROGRAM*   LOCAL

                                        ENTRY POINTS

                                           000001 FDIV

                                        *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                                        *        A FUNCTION CALLED WITH ONE ARGUMENT - FDIV RETURNS THAT ARG
                                        *        UNCHANGED, UNLESS IT IS ILLEGAL (IE - INFINITE OR INDEFINITE)
                                        *        IN WHICH CASE IT RETURNS ZERO.  TYPICAL USE IS ...
                                        *                 INSTEAD OF  X=A/B
                                        *                      WRITE X=FDIV(A/B)  IF B MIGHT BE ZERO
                                        *
                                                  ENTRY     FDIV
                    000000  0604112000000000001   VFD       24/6CFDIV,36/1
                    000001                FDIV    BSSZ      1
                    000002  54110               SA1       B1
                                10611           BX6       X1
                                    0351000004 *  OR        X1,SETZRO
                    000003  0361000001 *        DF        X1,FDIV
                    000004  76600         SETZRO  SX6     B0
                                0200000001 *    JP        FDIV
                                                END
                    000005

                            034513      UNUSED STORAGE          18 STATEMENTS       2  SYMBOLS
```

```
      SUBROUTINE FLAC (S,C,CP)
C     SOURCE DATE 69.0613   CORRECT CALC OF ALPHAC
C     SOURCE DATE 69.0328   RE-ARRANGE SETUP OF ANC, DUMMY(6-6), AND D ARRAY
C     SOURCE DATE 68.1205   USE KONVERG INSTEAD OF LIVERGE
C                           ALSO MAKE AMBIENT-PRESSURE-THRUST-CORRECTION
C                           DEPEND ON SONIC SPEED AS WELL AS DENSITY
C     SOURCE DATE 68.0717   USE QUADRATIC FORM FOR ALF OF CN1,CN2
C     SOURCE DATE 68.0311   PASS CP (ALSO) TO CAXIAL AND CNORML
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 68.0208    INCLUDE DRAG WHEN FINDING ALPHA
C     SOURCE DATE 68.0207    CORRECT ALF FOR NO-MNVR, NO-RESP FLIGHT
C     SOURCE DATE 67.0811   AMEND FORMULA FOR ACCEL WITH RESPONSE
C     SOURCE DATE 67.0726   CALL TRAIDERR IF ERROR CONDITION
C     SOURCE DATE 67.0707   CALL CAXIAL, CNORML FOR AERO COEFFS, ALSO
C           67.0707   ANC IS IN RADIANS IF -FLAG- IS NON-BLANK
C     SOURCE DATE  67.0612      ADAPTED FROM FLACON
C                                                                      191
C         COMPUTES ACCELERATION OF ROCKET WITH DAMPED SECOND-ORDER      192
C         RESPONSE TO NORMAL ACCELERATION COMMANDS                      193
C                                                                       194
C         S       -  CORRECT ROCKET STATE VECTOR, FOLLOWED BY CURRENT   195
C                    ROCKET ANGLE-OF-ATTACK VECTOR                      196
C         C       -  CHARACTERISTICS VECTOR FOR ROCKET TYPE             197
C              C(KSTAGE,1)  -  VACUUM THRUST AT IGNITION
C              C(KSTAGE,2)  -  NOZZLE EXIT AREA                  66.0601
C              C(KSTAGE,3)  -  TOTAL MASS AT STAGE IGNITION      66.0601
C              C(KSTAGE,4)  -  TOTAL MASS AT STAGE BURNOUT       66.0601
C              C(KSTAGE,5)  -  STAGE BURNING TIME                66.0601
C              C(KSTAGE,6)  -  REFERENCE AREA                    66.0601
C              C(KSTAGE,7)  -  SUBSONIC AXIAL FORCE COEF.        66.0601
C              C(KSTAGE,8)  -  SUPERSONIC AXIAL FORCE PMTR A     66.0601
C              C(KSTAGE,9)  -  SUPERSONIC AXIAL FORCE PMTR B     66.0601
C              C(KSTAGE,10) -  HYPERSONIC AXIAL FORCE COEF.      66.0601
C              C(KSTAGE,11) -  MAXIMUM ALLOWABLE NORMAL ACCELERATION 66.0601
C              C(KSTAGE,12) -  MAXIMUM ALLOWABLE ANGLE OF ATTACK 66.0601
C              C(KSTAGE,13) -  NORMAL FORCE PARAMETER CN1        66.0601
C              C(KSTAGE,14) -  NORMAL FORCE PARAMETER CN2        66.0601
C              C(KSTAGE,15) -  RESPONSE DAMPING FACTOR ZETA      66.0601
C              C(KSTAGE,16) -  RESPONSE UNDAMPED RESONANCE FREQUENCY 66.0601
C         CP      -  CHARACTERISTIC VECTOR FOR PARTICULAR ROCKET      210
C              CP(1)-CP(3) -  CURRENT COMMANDED ACCELERATION (NORMAL  211
C                             TO CURRENT VELOCITY)                    212
C              CP(4)       -  CURRENT STAGE NUMBER KSTAGE             213
C              CP(5)       -  PHASE WITHIN STAGE KPHASE               214
C                             KPHASE=-1 -  PRE-BURNING COAST          215
C                             KPHASE= 0 -  BURNING                    216
C                             KPHASE=+1 -  POST-BURNING COAST         217
C              CP(6)       -  TIME OF IGNITION OF CURRENT STAGE       218
C              CP(7)       -  LAUNCH TIME                             219
C              CP(8)       -  STAGE 1 IGNITION TIME                   220
C              CP(9)       -  STAGE 1 SEPARATION TIME                 221
C              CP(10)      -  STAGE 2 IGNITION TIME                   222
C              CP(11)      -  STAGE 2 SEPARATION TIME                 223
C              CP(12)      -  STAGE 3 IGNITION TIME                   224
C
C                                                                      225
C         SIGN OF C(KSTAGE,1) SELECTS MOTOR BURNING CHARACTERISTICS    226
```

PRECEDING PAGE BLANK-NOT FILMED

```
C               POSITIVE = CONSTANT BURNING RATE, CONSTANT THRUST
C               NEGATIVE = EXPONENTIAL BURNING RATE, CONSTANT
C                          THRUST-TO-MASS RATIO                                 229
C                                                                               230
C          RETURNS S(N)=S(10) AND CP(4)=CP(6).  THE LATTER ARE SET UP           231
C          TO AGREE WITH AND FACILITATE INTEGRATION BY SUBROUTINE STINT         232
C                                                                               233
C          IF OMEGA IS ZERO, ROCKET IS ASSUMED TO RESPOND INSTANTLY TO          234
C          COMMANDS.                                                            235
C          IF OMEGA IS NOT ZERO, SECOND-ORDER RESPONSE IS ASSUMED AND           236
C          S(11)=S(20) IS USED TO STORE AN ANGLE-OF-ATTACK VECTOR (THE          237
C          DIFFERENCE BETWEEN THE UNIT AXIS VECTOR AND UNIT VELOCITY            238
C          VECTOR) AND ITS RATES.  ACCELERATIONS OF THIS VECTOR ARE             239
C          RETURNED IN S(18)-S(20) FOR INTEGRATION BY STINT ALONG WITH S.       240
C                                                                               241
C                                                                               243
      CCOMMRG,HASCOM
000004        COMMON /HASCOM/
     1   PROGRM,    KPAGE,     LINE,      TOPDAY,    RUN,       RUNID(6),
     2   MSG,       FLAG,      DATE,      MAATM,     MAAMG,     MAALN,
     3   KLASS,     KGROUP,    KUNITS,    KSTINT,    KOUND,     IFEOF
     4  ,NOFILE
      CCOMMRG,HASCOM
C                                                                    09/15/69
      CCOMMRG,CONCOM
000008        COMMON /CONCOM/
     1   PI,        SND,       SLV,       SMF,       SKF,       RBODY,
     2   GACC,      GCON,      RMODY,     RMOZRO,    TWOPI,     HAFPI
      CCOMMRG,CONCOM
C                                                                    09/15/69
      CCOMMRG,INTCOM
00000A        COMMON /INTCOM/
     1   DT,        KSTEP,     NOIFEQ,    SDANDA(54)
00000E        DIMENSION  SD(12,4),  XINT(12,4)
00000A        EQUIVALENCE (SD(7),XINT(1),SDANDA(7))
      CCOMMRG,INTCOM
C                                                                    09/15/69
000006        DIMENSION S(20),C(3,16),CP(12),D(16),DUMMY(6),ANC(3),ACC(3),
     1 VC(3),G(3),O(3),WORK(N)                                          68.1205
000006        EQUIVALENCE (D(1),FT),(D(2),CAE),(D(3),XM1),(D(4),XM2),(D(5),TB),
     1 (D(6),SMEF),(D(7),CA),(D(8),A),(D(9),B),(D(10),CA2),             66.0601
     2 (D(11),ANM),(D(12),ALPHAM),(D(13),CN),(D(14),CN2),(D(15),ZETA),
     3 (D(16),OMEGA)
00000A        EQUIVALENCE (DUMMY(1),ANC(1)),(DUMMY(4),KSTAGE),(DUMMY(5),KPHASE),      250
     1 (DUMMY(6),TREF)                                                  251
00000A        DATA (BLANK=1H )                                         67.0707
C
000006        T=S(1)                                                        254
000007        CALL ARIT(3,CP,DUMMY)                                    69.032M
000011        IF (KSTEP.NE.0) GO TO 11                                      255
000014        T1=T+.5*DT                                                    256
000017        DO 4 N=7,12                                                   257
000021        IF(CP(N).GT.T1) GO TO 5                                  69.032M
000025    4   CONTINUE                                                      260
000026        N=13                                                          261
000027    5   KSTAGE=(N-6)/2                                                262
000032        TREF=CP(N-1)                                                  263
000036        KPHASE=MOD(N,2)-1                                             264
```

```
000041          IF (XPHASE.LT.0) GO TO 8                                    265
000042          IF (TT.LE.THEF+C(NSTAGE,5)) GO TO 8                         66.0601
000047          THEF=THEF+C(NSTAGE,5)                                       66.0601
000051          XPHASE=1                                                    266
000052        8 CALL AMI(S,C,XPHASE,CP(6))                                  69.0328
000056          GO TO (9,10),N                                              69.0328
000061       9  C(N)=C(NSTAGE,N)                                           273
000071          ALPHAM =AMI(I(ALPHAM,MAPMI)                                 69.0613
000074       11 IF (XSTAGE.EQ.0) RETURN                                     271
000076          RHO = DNSITY(S)                                             66.0601
000102          SOUND=SONIC(ALTF(S))                                        68.1205
              C --  CALCULATE AMBIENT PRESSURE IN NEWTON/SQ METER  OR  POUNDAL/SQ FT
000112          PRES=RHO*SOUND**2/1.401                                     68.1205
000114          IF (XPHASE.GT.0) GO TO 18                                   274
000117          IF (XPHASE.EQ.0) GO TO 15                                   275
000117          XMASS=XM1                                                   276
000121          GO TO 19                                                    277
000121       15 IF (FT1.LT.0.) GO TO 17                                     66.0601
000123          FT = FT1-CAE*PRES                                           68.1205
000125          XMASS=XM1-(XM1-XM2)*FDIV((T-THEF)/TB)                       68.0209
000136          GO TO 20                                                    281
000141       17 CEX=(XM2/XM1)**FDIV((T-THEF)/TB)                            68.0209
000153          FT=CEX*FT1-CAE*PRES                                         68.1205
000157          XMASS=CEX*XM1                                               284
000161          GO TO 20                                                    285
000163       18 XMASS=XM2                                                   286
000165       19 FT=0.                                                       287
000166       20 ANGMAG=AMAG(ANG)                                            288
000171          CALL GRAV(S,G)                                              290
000174          V=XMAG(S(5))
000203          Q=0.5*RHO*V**2
000205          QSOM=FDIV(SREF/XMASS)                                       68.0209
000213          QSOM=QSOM
000214          FOM=FDIV(FT/XMASS)
000220          VMACH=V/SOUND                                               68.1205
              C
000222          ASSIGN 400 TO MAKECN
000223          ASSIGN 420 TO MAKECA
000224          ASSIGN 440 TO MAKEYS
              C
000225          IF(ANGMAG.GT.0. .OR. OMEGA.GT.0.) GO TO 203
000235          ASSIGN 202 TO KUMBAK
000236          ALF=0.                                                      68.0207
000237          GO TO MAKECA
000243      202 AA=FDIV((FOM-CA*QSOM)/V)                                    68.0209
000252          CALL VECLIN(AA,S(5),1.,G,S(8))
000262          RETURN
              C
000263      203 ASSIGN 215 TO KUMBAK                                        68.0208
000264          ALF=ALPHAM
000266          GO TO MAKECA
000273      215 ASSIGN 216 TO KUMBAK                                        68.0208
000274          GO TO MAKECA                                                68.0208
000300      216 GLIMAMO=CSUBN*QSAM
000302          GLIMTHW=GLIMAMO*COS(ALF)+(FOM-QSOM*CA)*SIN(ALF)             68.0208
000315          IF(ANGMAG.GT.0.) GO TO 210
              C
```

```
000322      214 ALFA=0.
000323          ASSIGN 233 TO KUMBAR
000324          GO TO MAKUVS
                C
000330      210 LCOA=COS(ALF)
000332          ALF=ALPHAM
000333          IF(GLIMAPL.LT.ANF) GO TO 220
000336          CGOAL=ANM/GSNM
000337          ASSIGN 214 TO KUMBAR
000340          OREMM=.002                                              68.1205
000342          CALL KONVSET(.01,OREMM,OMK)                             68.0204
000346          ALF=FLIV(CGOAL/UCOA)
000351          GO TO MAKECA
000356      215 TEST=CSUMM-CGOAL
000360          IF(KONVERG(TEST,ALF,OMK)) 219,399,220                   68.1205
000367      219 CALL THA DERR(24HALPHMA TROUBLE IN FLAC =)               68.0209
000371          RETURN
                C
000372      220 ALPHMA=ALF                                              67.0707
000374          IF(FLAG.EQ.BLANK) GO TO 222                             67.0707
000377          ALF=ANCMAG                                              67.0707
000401          GO TO 230                                               67.0707
000401      222 ALF=ALPHAM
000403          IF(GLIMTMM.LT.ANCMAG) GO TO 230
000405          ASSIGN 224 TO KUMBAR
000407          CTMY=FLIV(ANCMAG/(USNM*FLIV(FUM/UCOA)))                 68.0209
000417          OREMM=.001                                              64.0613
000420          CALL KONVSET(.01,OREMM,OMK)                             66.1205
000423          ALF=FLIV(CTMY/UCOA)                                     64.0204
000430          IF(CN2.NE.0.) ALF =                                     67.0717
               *    (-CN)+SQRT((CN)**2+4.*CN2*CTRY))/(2.*CN2)          67.0717
000465          GO TO MAKECA
000467      224 ASSIGN 226 TO KUMBAR                                    68.020H
000473          GO TO MAKECA                                            68.020H
000476      226 TEST=CSUMM*GSNM*COS(ALF)+(FUM-USUM*CA)*SIN(ALF)-ANCMAG  68.020H
000560          ASSIGN 224 TO KUMBAR                                    68.020H
000575          IF(KONVERG(TEST,ALF,OMK)) 225,399,230                   68.1205
000604      225 CALL THA DERR(24HALPHAC TROUBLE IN FLAC =)               64.0209
000610          RETURN
                C
000507      230 ALPHAC=ALF
000511          ALFA=AMIN1(ALPHAC,ALPHMA,ALPHAM)
000516          ASSIGN 232 TO KUMBAR
000517          GO TO MAKUVS
000525      232 CSA=COS(ALFHA)
000530          SNA=SIN(ALFHA)
                C
000532      234 ALF=ALPHA
000534          ASSIGN 236 TO KUMBAR
000535          GO TO MAKECA
000542      236 AX=FUM-CA*USUM
000545          ASSIGN 236 TO KUMBAR
000546          GO TO MAKECA
000552      236 AN=CSUAN*GSNM
000554          IF(ANCMAG.GT.0.) GO TO 240
000557          CON1=OMEGA**2
000560          CON2=2.*OMEGA*ZETA
000562          DO 238 N=1,3
```

```
000563          S(N+17)=CON1*(VC(N)-U(N))-CON2*S(N+14)                          68.0209
000572          U(N)=FDIV(S(N+4)/UMAG)
000577    230   S(N+7)=AA*U(N)+G(N)
000610          RETURN
                C
000610    240   IF(OMEGA.NE.0.) GO TO 36                                        322
000611          DO 34 N=1,3                                                     323
000613    34    S(N+7)=AA*(VU(N)*CSA+ACU(N)*SNA)+AN*(ACU(N)*CSA-VU(N)*SNA)+G(N)  324
000652          RETURN
                C                                                             325
000653    36    CSF=COS(ALFA)                                                   326
000636          SNF=SIN(ALFA)                                                   327
000640          CON1=OMEGA**2                                                   328
000641          CON2=2.*OMEGA*ZETA                                            68.0209
000643          CON3=FDIV(AN/SNA)                                               330
000650          DO 38 N=1,3                                                     331
000653          S(N+17)=CON1*(CSF*VU(N)+SNF*ACU(N)-U(N))-CON2*S(N+14)         68.0209
000666          U(N)=FDIV(U(N)/UMAG)                                            332
000673    38    S(N+7)=AA*U(N)+CON3*(U(N)*CSA-VU(N))+G(N)                       333
000711          RETURN
                C                                                             68.1205
000711      399 GO TO MARECA
                C MARECIN
000715    400   CSUBN=CINEML(VMACH,ALF,C,KSTAGE,CP)                          68.0311
000725          GO TO KUMMAR                                                  67.0707
                C
                C MARECA
000730    420   CA=CAXIAL(VMACH,ALF,C,KSTAGE,CP)                              68.0311
000740          GO TO KUMMAR                                                  67.0707
                C
                C MARUVS
000743    440   DO 442 N=1,3                                                  68.0209
000745          VU(N)=FDIV(S(N+4)/V)                                            314
000755          U(N)=S(N+11)+VU(N)                                            68.0209
000761    442   ACU(N)=FDIV(ANC(N)/ANCMAG)                                    67.0811
000773          UMAG=AMAG(U)
000776          ALPHA=ALFA
000777          IF(OMEGA.NE.0.) ALPHA=SEPA(U,VU)
001005          GO TO KUMMAR
                C
                C
001013          END  MELAC
```

```
      SUBROUTINE FLIER(S,CP,KONDA,DTE)                                  335
C     SOURCE DATE  68.1205   EXTEND REVISION OF 68.0507
C     SOURCE DATE  68.0612    DUN=1 JUMP TO S1, 8 IF ANC UNDEFINED
C     SOURCE DATE  68.0507   END MANEUVER STAGES ON ZERO IN CP ARRAY
C     SOURCE DATE 67.0823   FIX FOR ROUND EARTH
C     SOURCE DATE 66.0801   EXPANDED GOAL CAPABILITY
C                                                                       336
C         GENERATES MANEUVER COMMANDS FOR FLIGHT PATH PROGRAM           337
C                                                                       338
C         S      - VEHICLE STATE VECTOR                               339
C         CP     - PARTICULAR VEHICLE CHARACTERISTICS                 340
C              CP(1)-CP(12)   - SEE #FLAC# FOR DEFINITIONS             341
C              CP(13)-CP(15)  - MANEUVER STATUS INDICATORS             342
C                   CP(13)    - NUMBER OF CURRENT MANEUVER (ZERO IF    343
C                                     NO FLIGHT PROGRAM IS IN USE)      344
C                   CP(14)    - VALUE OF MANEUVER GOAL AS LAST TIME    345
C                               STEP BEGAN                              346
C                   CP(15)    - TIME AT WHICH CURRENT MANEUVER BEGAN   347
C              CP(16)-CP(20)  - SPECIFICATION FOR FIRST MANEUVER       348
C                               IN FLIGHT PROGRAM                       349
C                   CP(16)    - MANEUVER ACCELERATION MAGNITUDE        350
C                   CP(17)    - MANEUVER DIRECTION (MEASURED FROM      351
C                               HORIZONTAL, ABOUT VELOCITY VECTOR)      352
C                               ZERO IS HORIZONTAL TO THE RIGHT,
C                               +90 DEG IS DOWN.
C                   CP(18)    - RATE OF CHANGE OF MANEUVER DIRECTION   353
C                   CP(19)    - MANEUVER GOAL (VALUE OF CRITERION AT   354
C                               WHICH MANEUVER IS TO END)               355
C                   CP(20)    - CODE DEFINING GOAL (SEE FOAL FOR DEF.) 356
C              CP(21)-CP(25)  - MANEUVER 2 DEFINITION                  357
C              ETC.             ETC.                                   358
C         DTE    - ESTIMATED TIME STEP TO BRING VEHICLE TO GOAL        359
C         KONDA  - CURRENT ITERATION STATUS IN ENDING MANEUVER         360
C                  (SEE #DISCON# FOR DEFINITIONS OF DTE AND KONDA)     361
C                                                                       362
      COMMON/#ASCON
000007        COMMON /#ASCON/
              1   PROGRM,    KPAGE,    LINE,     TODAY,    RUN,      RUNID(6),
              2   MSG,       FLAG,     DATE,     MAATM,    MAAPG,    MAALN,
              3   KLASS,     KGROUP,   KUNITS,   KSIINI,   KOUMD,    IFEOF
              4   ,#ADFILE
      COMMON/#ASCON
C                                                                 09/15/69
      COMMON/INTCON
000007        COMMON /INTCON/
              1   DT,        KSTEP,    NUIFEU,   SOANDA(56)
000007        DIMENSION  SO(12,4),  XINT(12,4)
000007        EQUIVALENCE (SO(7),XINT(1),SOANDX(7))
      COMMON/INTCON
C                                                                 09/15/69
C                                                                       364
000007        DIMENSION S(10),CP(21),M(3),U(3),ANC(3),Z(3)            67.0823
000007        DATA (Z=0.0,1.)                                         67.0823
000007        EQUIVALENCE (AE6,IEU)                                   66.0801
C
000007        MAVM=CP(13)*5.+1.                                       368
000012        IF (KSTEP.EQ.0) GO TO 2                                 369
```

PRECEDING PAGE BLANK-NOT FILMED

```
000013          AEL=CP(MNVR+4)                                          66.0601
000015          IF(IFW.EQ.0) GO TO 12                                   68.1205
000016          GOAL=CP(MNVR+3)                                             370
000020          IF(IFW.EQ.HHTIME INT) GOAL=GOAL+CP(15)
000024          CALL DISCON(GOAL+FOAL(S,CP(MNVR+4)),CP(14),KONDX,DT,DTE)
000041          IF (KONDX.GE.0) GO TO 2                                     373
000045          MNVR=MNVR+5                                                 374
000047          CP(13)=(MNVR-11)/5
000053          IF(CP(MNVR+4).EQ.0.) GO TO 11                           68.1205
000055          CP(14)=FOAL(S,CP(MNVR+4))
000083          CP(15)=S(1)                                                 377
000085        2 IF (CP(MNVR).EQ.0.) GO TO 11                            68.1205
000087          IF(KONDX.EQ.2) GO TO 5                                  67.0823
000071          CALL CROSS(S(5),Z,H)                                    67.0823
000074        3 IF(H(1).EQ.0. .AND. H(2).EQ.0. .AND. H(3).EQ.0.) H(1)=1.  67.0823
000105          GO TO 7                                                 67.0823
000110        5 CALL CROSS(S(5),S(2),H)                                 67.0823
000113          IF(H(1).NE.0. .OR. H(2).NE.0. .OR. H(3).NE.0.) GO TO 7  67.0823
000126          CALL CROSS(Z,S(2),H)                                    67.0823
000131          GO TO 3                                                 67.0823
000134        7 CALL CROSS(S(5),H,U)                                    67.0823
000137          PSI=CP(MNVR+1)+(S(1)-CP(15))*CP(MNVR+2)                     378
000147          CALL VECLIN (CP(MNVR)*COS(PSI),H,                       68.1205
               *                CP(MNVR)*SIN(PSI),U,ANC)                68.1205
000171          X=1.                                                        379
000173          IF (DOT(ANC,CP).LT.0. .AND. KONDX.GE.0) X=-1.           68.1205
000212        8 DO 9 I=1,3                                                  380
000214        9 CP(I)=X*ANC(I)                                              381
000221          RETURN                                                      382
      C
000222       12 CP(13)=0.                                               68.0507
000223          KONDX=0                                                 68.1205
000224       11 CALL AMIT(-3.0,CP)                                      68.0612
000227          RETURN                                                  68.0612
000230          END  FELTER
```

```
      SUBROUTINE FLIGHT(AXEL,GUIDE,ST,C,CP,TSTEP,GOALFN,VAL,KOD,INS,NST)
C     SOURCE DATE 69.0909    RETURN UPON REACHING INPUT LIMIT NST
C     SOURCE DATE 69.0709    USE WRITE STATEMENT INSTEAD OF CALL TO WRITIT
C     SOURCE DATE 69.0328    FIX SAVING OF ACCELERATION VECTOR
C     SOURCE DATE 69.0109    BRAND NEW CODE
C                                                                        402
C         INTEGRATES A GUIDED TRAJECTORY TO A SPECIFIED END CONDITION    403
C                                                                        404
C         AXEL   -  SUBROUTINE FOR CALCULATING TRAJECTORY ACCELERATIONS
C         GUIDE  -  SUBROUTINE FOR CALCULATING GUIDANCE COMMANDS         406
C         ST     -  ARRAY IN WHICH INITIAL TRAJECTORY STATE MUST BE
C                   FIRST ENTRY AND FINAL STATE PRODUCED BY FLIGHT WILL  408
C                   BE LAST ENTRY                                        409
C         C,CP   -  ARGUMENTS OF SUBROUTINES AXEL AND GUIDE
C         TSTEP  -  TIME STEP FOR INTEGRATION
C         GOALFN -  FUNCTION RETURNING VALUE OF CRITERION FOR ENDING     412
C                   TRAJECTORY INTEGRATION                               413
C         VAL    -  NUMERICAL VALUE OF CRITERION AT WHICH INTEGRATION
C                   IS TO BE TERMINATED                                  415
C         KOD    -  ARGUMENT FOR GOALFN
C         INS    -  CONTROLS INSERTION OF INTERMEDIATE VECTORS INTO
C                   ST ARRAY  ...
C                    INS=0     GETS FIRST AND LAST VECTORS ONLY
C                    INS=-N    GETS VECTORS EVERY N TIME STEPS
C                    INS=N     GETS VECTORS EVERY N TIME STEPS PLUS AT
C                              EVERY GUIDANCE AND STAGING DISCONTINUITY
C         NST    -  NUMBER OF FINAL STATE RETURNED IN ARRAY STATES
C         ***  IMPORTANT     - BEFORE CALLING FLIGHT, SET NST TO          69.0909
C                              THE HIGHEST NUMBER OF VECTORS THE ST
C                              ARRAY CAN HOLD. FLIGHT WILL PRINT A
C                              MESSAGE AND RETURN IF YOU HIT THIS LIMIT  69.0909
C                                                                        422
      COMPKG,INTCON
000016        COMMON /INTCON/
     1   DT,       KSTEP,    NDIFEQ,   SOANDA(54)
000016        DIMENSION  SO(12,4),  XINT(12,4)
000016        EQUIVALENCE (SO(7),XINT(1),SOANDA(7))
      COMPKG,INTCON
C                                                                  09/15/69
      COMPKG,MULCON
000016        COMMON /MULCON/
     1   SMULS(10,4)
      COMPKG,MULCON
C                                                                  09/15/69
000016        DIMENSION S(10,4)
000016        EQUIVALENCE (S,SMULS)  $  DATA (S=40(0))
000016        DIMENSION ST(10,NST),C(3,16),CP(20),SAVE(3,4)
C                                                                        424
C - - - - - - - - - - - - - - - - - -   INITIALIZE
000016        CALL XMIT(3,DT,SAVE(1,3))
000022        DT=DTE=TSTEP
000030        KSTEP=0
000031        NDIFEQ=1
000032        IF(C(1,1A).NE.0. .OR. C(2,16).NE.0. .OR. C(3,16).NE.0.) NDIFEQ=2
000044        GOAL=VAL  $  IF(KOD.EQ.8HTIME INT) GOAL=GOAL+ST(1,1)
000057        INA=IABS(INS)
000054        MAX=NST-1-NDIFEQ                                           69.0909
```

```
000057          KAE=KAM=KAS=0                                                   69.0709
000062          GLAS=GOALFN(ST,KOD)                                             69.0709
000075          CALL DISCON(GOAL,GLAS,GLAS,KAE,DT,DTE)                          69.0709
000101          NST=1                                                           69.0709
000103          IF(KAE.NE.0) GO TO 306                                          69.0709
000107          NST=2                                                           69.0709
000110          GLST=TFRS=ST(1,1)
000113          CALL APIT(10*NDIFEQ,ST,S)
000117          LSTP=1  $  L=6
000121          IF(TSTEP.GT.0.) GO TO 22
000127          LSTP=-1  $  L=13
000131       22 ASSIGN 24 TO LBAK  $  GO TO 450
000133       24 ASSIGN 110 TO LBAK
000136          IF(DT/TLEFT.GT.1.) DT=TLEFT                                     69.0328
000141          CP(1)=CP(2)=CP(3)=0.
000147          IF(CP(13).NE.0.) CALL GUIDE(S,CP,KAM,DTM)
000164          CALL AXEL(S,C,CP)
000177          ASSIGN 90 TO KUMBAK  $  GO TO 400
                C
                C - - - - - - - - - - - - - - - - - -  INTEGRATE TRAJECTORY
000200       90 DTM=DTS=TSTEP
000202      100 CALL STINT
000203          LAM=KAM                                                         69.0328
000205          IF(KSTEP.NE.0) GO TO 102
000211          SAVE(1,1)=S(8,1)  $  SAVE(2,1)=S(9,1)  $  SAVE(3,1)=S(10,1)
000216          IF(NDIFEQ.EQ.1) GO TO 102
000220          SAVE(1,2)=S(8,2)  $  SAVE(2,2)=S(9,2)  $  SAVE(3,2)=S(10,2)
000224      102 IF(CP(13).NE.0.) CALL GUIDE(S,CP,KAM,DTM)
000241          CALL AXEL(S,C,CP)
000254          IF(KSTEP.NE.0) GO TO 100
000255          TLEFT=TLEFT-DT                                                  69.0328
                C
                C - - - - - - - - - - - - - - - - - -  CHECK FOR END CONDITIONS
000257          CALL DISCON(GOAL,GOALFN(S,KOD),GLAS,KAE,DT,DTE)
000276          IF(L.LE.12.AND.L.GE.7)
               .   CALL DISCON(CP(L),S(1,1),GLST,KAS,DT,DTS)
000326          NSTEP=(S(1,1)-TFRS)/TSTEP + .001
000336          IF(NSTEP.GT.10000 .OR. KAE+KAM+KAS.GT.30) GO TO 500
                C
000350          IF(KAE) 300,104,200
000352      104 IF(KAM.NE.0 .OR. KAS.NE.0) GO TO 190                            69.0328
000360          DT=TSTEP
000361          IF(DT/TLEFT.LT.1.) GO TO 106                                    69.0328
000364          DT=TLEFT                                                        69.0328
000365          CALL AXEL(S,C,CP)                                               69.0328
000377      106 IF(INS.EQ.0) GO TO 90
000401          IF(MOD(NSTEP,INA).NE.0) GO TO 90
000405          ASSIGN 90  TO KUMBAK  $  GO TO 400
                C
                C - - - - - - - - - - - - - - - - - -  END OF STAGE OR MANEUVER
000408      110 IF(INS.LE.0) GO TO 210
000410          S(8,1)=SAVE(1,4)  $  S(9,1)=SAVE(2,4)  $  S(10,1)=SAVE(3,4)
000415          ASSIGN 112 TO KUMBAK  $  GO TO 400
000418      112 DT=0.01*TSTEP
000420          CALL AXEL(S,C,CP)
000437          ASSIGN 210 TO KUMBAK  $  GO TO 400
                C
                C - - - - - - - - - - - - - - - - - -  FLEW PAST DISCONTINUITY
```

```
000433      190 IF(RXS.GT.1 .OR. LAM.GT.0) GO TO 200                          69.0328
000443          SAVE(1,4)=SAVE(1)  $  SAVE(2,4)=SAVE(2)  $  SAVE(3,4)=SAVE(3)
000447      200 IF(RXS.LT.0) GO TO 450                                        69.0328
000451          IF(RXM.LT.0) GO TO 110                                        69.0328
000452          S(8,1)=SAVE(1,1)  $  S(9,1)=SAVE(2,1)  $  S(10,1)=SAVE(3,1)   69.0328
000457          IF(NDIFEQ.EQ.1) GO TO 202
000461          S(8,2)=SAVE(1,2)  $  S(9,2)=SAVE(2,2)  $  S(10,2)=SAVE(3,2)
000465      202 IF(TSTEP.LT.0.) GO TO 204
000467          DT=AMIN1(DTE,DTM,DTS)
000474          GO TO 90
000475      204 DT=AMAX1(DTE,DTM,DTS)
000503          GO TO 90
                C
                C - - - - - - - - - - - - - - - - - -   GET BACK IN PHASE WITH TSTEP
000504      210 DT=(FNS*(NSTEP+1)*TSTEP - S(1,1)
000513          CALL AXEL(S,C,CP)
000525          GO TO 90
                C
                C - - - - - - - - - - - - - - - - - -   END OF TRAJECTORY
000525      300 ASSIGN 302 TO KOMBAK  $  GO TO 400
000527      302 IF(NDIFEQ.LT.2 .OR. INS.EQ.0) GO TO 306                       69.0328
000537          CALL AMIT(10,S(1,2),S(1,1))                                   69.0328
000545          ASSIGN 304 TO KOMBAK  $  GO TO 400                            69.0328
000552      304 NST=NST-1                                                     69.0328
000554      306 CALL AMIT(3,SAVE(1,3),DT)                                     69.0328
000561          RETURN
                C
                C - - - - - - - - - - - - - - - - - -   INSERT VECTOR INTO ST ARRAY
000562      400 NST=NST+1
000564          CALL AMIT(10,S,ST(1,NST))
000576          IF(NST.NE.MAX) GO TO KOMBAK                                   69.0709
000610          CALL COUNOUT(3)                                               69.0709
000612          WRITE(6,401) NST                                              69.0709
000620      401 FORMAT(*0* 30A *= FLIGHT HAS COMPUTED*I6* STATES =*/* *)      69.0709
000626          GO TO 302                                                     69.0909
                C
                C - - - - - - - - - - - - - - - - - -   SEARCH FOR STAGING TIME
000624      450 L=L+LSTP
000626          TLEFT=1.E8                                                    69.0328
000627          IF(L.LT.7 .OR. L.GT.12) GO TO 452
000636          TLEFT=CP(L)-S(1)                                              69.0328
000640          IF(TLEFT/TSTEP.LT. .001) GO TO 450                            69.0328
000643      452 KAS=0
000644          GO TO LBAK
                C
                C - - - - - - - - - - - - - - - - - -   ERROR MESSAGE
000650      500 CALL HEAD(10)
000652          CALL STOUT(4)*= FLIGHT HAS FAILED TO REACH A SOLUTION =*
                .              10111),X,S,NDIFEQ)
000666          CALL LSKIP(3)
000670          CALL COUNOUT(3)
000702          WRITE(6,501) KAM,DTM,CP(13),
                .                KAS,DTS,CP(L),GLST,
                .                KXE,DTE,GOAL,GLAS
000726      501 FORMAT(* END MANUVR FLAG = *I4*   DT = *F6.3*   MANEUVER = *F6/
                .       * END STAGE  FLAG = *I4*   DT = *F6.3*   GOAL = *F11.3,
                .         *   LASTGOAL = *F11.3/
                .       * END FLIGHT FLAG = *I4*   DT = *F6.3*   GOAL = *F11.3,

                .         *   LASTGOAL = *F11.3)
000770          CALL TRAILER(IM )
000740          GO TO 302
000734          END PFLIGHT
```

```
      SUBROUTINE FLINITITLE,MODE,C,CP)                                    507
C     SOURCE DATE 69.0718    INCREASE NUMBER OF UNIT-CONVERSION OPTIONS
C     SOURCE DATE 68.1205    CORRECT TWO-MMANCH COMPUTED GO TO
C                            ALSO MAKE AMBIENT-PRESSURE-THRUST-CORRECTION
C                            DEPEND ON SONIC SPEED AS WELL AS DENSITY
C     SOURCE DATE 68.0828    CORRECT NOZZLE-AREA SCALING
C     SOURCE DATE 68.0826     ZERO OUT NEXT-UNSPECIFIED MANEUVER PHASE
C     SOURCE DATE 68.0311    CLEAN UP DRAG PARAMETERS
C     SOURCE DATE 68.0301    CHANGE SPELLING OF UNIT -NMI-
*     SOURCE DATE 68.0209        CONVERT TO CDC 6400
C     SOURCE DATE  67.0912     FIX INPUT FOR M1 = 3
C     SOURCE DATE  67.0824    USE CP(7) FOR LAUNCH TIME
C     SOURCE DATE 67.0726    CALL OLDATA ON READING SPECIAL DATA-CARD
C     SOURCE DATE 67.0707    ALLOW OPTION  M2=5, ALSO MANEUVER IS IN
C                            DEGREES IF -FLAG- IS NON-BLANK
C     SOURCE DATE 67.0203      FOR M1=1, STORE INPUT MASS IN C(N,4)
C     SOURCE DATE 66.1222    RE-ARRANGE OUTPUT, ALLOW MANY STAGES MNVR
C     SOURCE DATE 66.0601    ALLOW DRAG PROFILE TO BE INPUT
C                                                                         508
C         READS, PRINTS, AND SCALES FLIGHT PARAMETERS                     509
C                                                                         510
C         TITLE   -  HOLLERITH TITLE TO PRECEDE OUTPUT                    511
C         MODE    -  CONTROL WORD OF 3 DIGITS M1,M2,M3 WHICH GOVERN READ-
C                    IN OF THRUST-MASS PARAMETERS, AERODYNAMIC PARA-
C                    METERS, AND MANEUVER SCHEDULE, RESPECTIVELY, AS
C                    FOLLOWS -
C
C         VALUE      PARAMETERS READ
C         -----      ---------------------------------------------------
C         M1=0     - (NONE)
C            1       MASS, POUNDS (FLIGHT IS UNPROPELLED)
C            2       VACUUM THRUST, POUNDS (CONSTANT-THRUST PROPULSION IS
C                          ORDINARILY USED.  ALTERNATIVELY, CONSTANT-
C                          ACCELERATION PROPULSION MAY BE HAD BY MARKING
C                          THIS CARD IN COLS 1-30 INITIAL VACUUM THRUST
C                          ACC.,GS, AND ENTERING APPROPRIATE VALUES.)
C                    NOZZLE EXIT AREA, IN SQ
C                    INITIAL MASS, POUNDS
C                    FINAL MASS, POUNDS
C                    BURNING TIME, SECONDS
C                          (LAUNCH AND FIRST STAGE ASSUMED TO
C                          OCCUR AT TIME ZERO, WITH EACH STAGE
C                          SEPARATING AND IGNITING AS ITS
C                          PREDECESSOR BURNS OUT)
C               3     - AS FOR M1=2 PLUS EXPLICIT INPUT OF BURNING HISTORY,
C                    LAUNCH/SEPARATION TIME, SEC
C                    IGNITION TIME, SEC
C
C         M2=0     - (NONE)
C            1       REFERENCE AREA, FT SQ
C                    SUBSONIC AXIAL FORCE COEFFICIENT CX (AT MACH 0.5)     66.0601
C               2     - AS FOR M1=1 PLUS CX VARIATION WITH SPEED,          66.0601
C                    CX AT MACH 1    (ON THE HIGH SIDE)                    66.0601
C                    CX AT MACH 3                                          66.0601
C                    CX AT MACH 10                                         66.0601
C               3     - AS FOR M2=2 PLUS MANEUVERING PARAMETERS,
C                    MAX. NORMAL ACCELERATION, GRAVITIES
```

PRECEDING PAGE BLANK-NOT FILMED

```
C                    MAX. ANGLE OF ATTACK, DEG.
C                    NORMAL FORCE COEFFICIENT CN1, PER DEG
C                    NORMAL FORCE COEFFICIENT CN2, PER DEG SQ
C              4     - AS FOR M2=3 PLUS MANEUVER RESPONSE TO COMMANDS.
C                    RESPONSE DAMPING FACTOR
C                    RESPONSE UNDAMPED NATURAL FREQUENCY, CPS (IF ZERO,
C                         PRODUCES INSTANTANEOUS RESPONSE)
C              5     - AS FOR M2=4 PLUS ADDITIONAL DATA FOR                    67.0707
C                         EACH STAGE (EG, FOR AERO COEFFS).                  67.0707
C                         THE FIRST OF THESE CARDS MUST BE                   67.0707
C                         PRECEDED BY AN -IN1- CARD TELLING                  67.0707
C                         HOW MANY ADDITIONAL CARDS FOLLOW.                  67.0707
C
C          M3=0     - (NONE)
C             1     - MANEUVER SCHEDULE READ THROUGH INDICATED FINAL CARD
C                         WHICH MUST HAVE A MARK IN COL 71-72
C
C          ALL INPUT PARAMETERS EXCEPT MANEUVER SCHEDULES ARE READ FROM
C          TYPE 1 INPUT CARDS. MANEUVER SCHEDULES ARE READ FROM TYPE 2
C          CARDS. CARD FORMATS ARE AS FOLLOWS -
C                                                                            529
C                    COLUMN     FORMAT     CONTENTS                         530
C                    ------     ------     --------                         531
C                                                                            532
C       TYPE 1       1-40       5A8        PARAMETER IDENTIFICATION         66.0601
C                    41-70      3F10.0     PARAMETER VALUES FOR EACH STAGE    534
C                                                                            535
C       TYPE 2       1-10       A8,A2      CARD OR MANEUVER IDENTIFICATION    536
C                    11-40      3F10.0     MANEUVER AMPLITUDE, DIRECTION,     537
C                                          DIRECTION RATE                     538
C                    41-50      A8,A2      CRITERION FOR TERMINATING MANEUVER
C                    51-60      A8,A2      UNITS OF CRITERION FOR TERMINATION
C                    61-70      F10.0      NUMERICAL VALUE OF CRITERION AT WHICH  540
C                                          MANEUVER IS TO BE TERMINATED       541
C                    71-72      A2         FLAG INDICATING LAST MANEUVER CARD 542
C                    *** NOTE   -END DATA- IN COL 1-8 OF DATA-CARD
C                                          ALSO SERVES TO TERMINATE READING
C                                                                            543
C          ALLOWABLE MANEUVER TERMINATION CRITERIA ARE TABULATED IN
C          FUNCTION FUAL, WHICH EVENTUALLY MUST RECOGNIZE THEM.  IF THE
C          UNITS FOR THE CRITERION ARE (OR BEGIN WITH) KM, DEG, OR NMI,
C          SUBROUTINE FLIN WILL SCALE THEM TO METERS, RADIANS, AND FEET.
C                                                                            557
      CCOMPKG,BASCON
000007        COMMON /BASCON/
             1   PROGRM,    KPAGE,    LINE,     TOFDAY,   RUN,      RUNID(6),
             2   MSG,       FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
             3   KLASS,     KGROUP,   KUNITS,   KSTINT,   KOUND,    IFEOF
             4  ,NOFILE
      CCOMPKG,BASCON
      C                                                                  09/15/69
      CCOMPKG,CONCON
000007        COMMON /CONCON/
             1   PI,        SRD,      SLV,      SMF,      SKP,      RBODY,
             2   GACC,      GCON,     WRODY,    RHOZRO,   TWOPI,    HAFPI
      CCOMPKG,CONCON
      C                                                                  09/15/69
      CCOMPKG,CONVRT                                                       69.0718
```

```
000007          COMMON /CONVRT/ NNV, NAMVAL(3,11)                                69.0709
000007          DATA (NNV=11)                                                    69.0709
000007          DATA (NAMVAL =                                                   68.0527
               1    3HDEG,  2(.01745329252),                                    68.0527
               2    2HFT,   0.30480006,  1.0,                                   68.0527
               3    3HPSF,  4.88024,  1.0,                                      68.0527
               4    2HKM,   1000.,  3280.833,                                   68.0527
               5    3HNMI,  1853.25,  6080.3,                                   68.0527
               6    2HLB,   0.453592,  1.0,                                     68.0527
               7    1HG,    9.80665,  32.17398,                                 68.0527
               8    3HKFT,  304.80006,  1000.,                                  68.0527
               9    1HM,    1.0,   3.280833 ,                                   69.0709
               0    3HMRA,  2(.001) ,                                           69.0709
               1    3HSEC,  2(1.)  )                                            69.0709
          CCOMPKG,CONVRT                                                        69.0718
          C                                                                    09/15/69
000007          DIMENSION C(3,16),CP( 1),M(82),ID(55),PMTR(10),SKED(6),          69.0718
               *                        KARDS(4),D(3,11)                       69.0718
000007          DATA ((KARDS(I),I=1,4)=2,5,9,11),(BLANK=1H )                     69.0718
000007          EQUIVALENCE (AEG,IEQ)
          C
000007          M3=MOD(MODE,10)                                                    582
000013          M2=MOD(MODE/10,10)                                                 583
000020          MM=MIN0(M2,4)                                                   67.0707
000023          M1=MOD(MODE/100,10)                                                584
000031          IF(M1.EQ.0 .AND. M2.EQ.0) GO TO 30
000036          CALL OUTSET(454,12H     STAGE    ,12H      ONE     ,12H      ,H)
000041          CALL OUTSET( 74,12H     STAGE    ,12H      TWO     ,12H      ,H)      603
000045          CALL OUTSET( 94,12H     STAGE    ,12H     THREE    ,12H      ,H)      604
000051          IF (M1.EQ.0) GO TO 20
000055          KK=1  $  NLIN=5  $  IF(M1.GT.1) GO TO 12                        67.0912
000062          KK=3  $  NLIN=1                                                 67.0726
000064     12   CALL INCOL(1H*,M,ID,D,-NLIN)                                    67.0726
000071          IF(NOFILE.GT.0) RETURN
000073          IF(ID(1).EQ.RHOLD DATA) GO TO 40                                67.0726
000100          CALL XMIT(-15.0,C)                                              67.0726
000102          CALL XMIT(3*NLIN,D,C(1,KK))                                     67.0726
000114          IF(LINE+2+NLIN.GT.KARDS(MM),GT.MAXLN) CALL HEAD(1)              67.0726
000126          CALL OUTCOL(TITLE,M,X,X,0)                                      67.0707
000134          CALL OUTCOL(X,M,ID,C(1,KK),-NLIN)                               67.0726
000147          IF(M1.EQ.1) GO TO 145                                           68.1205
000154     125  DO 14  N=1,3
000156          IF (ID(1).NE.RHINITIAL ) GO TO 13
000160          C(N,1)=-C(N,1)*C(N,3)
000163     13   C(N,1)=GACC*C(N,1)
000166          C(N,2)=C(N,2)/144.                                              68.1205
000170          IF (KUNITS.NE.0) GO TO 14
000171          C(N,1)=C(N,1)/SKP
000173          C(N,2)=C(N,2)/SKP**2                                            68.1205
000175          C(N,3)=C(N,3)/SKP
000177          C(N,4)=C(N,4)/SKP
000201     14   CONTINUE
000203          GO TO 15                                                        67.0203
000204     145  DO 146 N=1,3                                                    67.0203
000206          IF(KUNITS.EQ.0) C(N,3)=C(N,3)/SKP                               67.0203
000211     146  C(N,4)=C(N,3)                                                   67.0203
000216     15   T=CP(7)                                                         67.0824
000220          DO 16 N=1,3
```

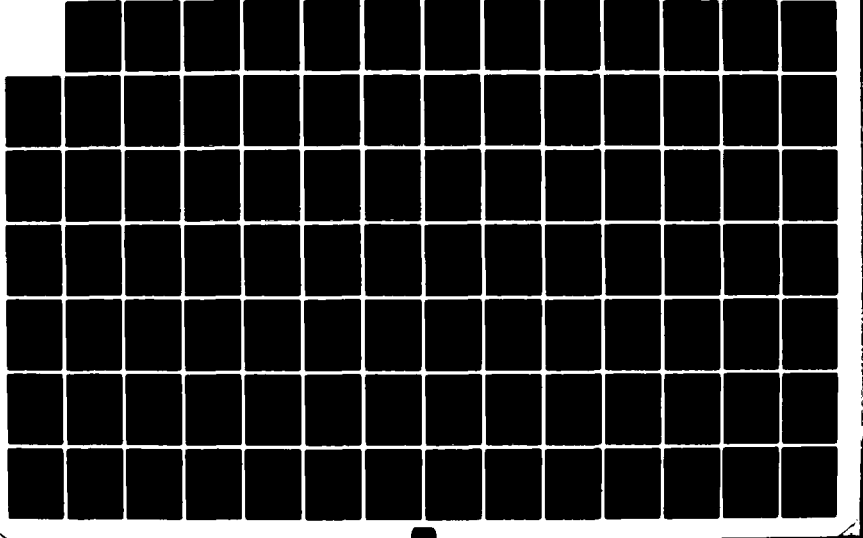
```
000221          SKED(N)=T
000223          SKED(N+3)=T
000224          T=T+C(N+5)
000226          IF (C(N+1+3).EQ.0.)T=1000000.
000231     16   CONTINUE
000233          IF(M1.LT.3) GO TO 17
000236          CALL INCOL(X,M,ID,SKED,-2)
000241          IF(NOFILE.GT.0) RETURN
000243          DO 165 N=1,6
000247     165  SKED(N)=SKED(N)+CP(7)
000253     17   DO 18  N=1,3
000255          CP(2*N+5)=SKED(N)
000260     18   CP(2*N+6)=SKED(N+3)
           C
000264     20   IF (M2.EQ.0) GO TO 30
000265          CALL XMIT(-33.0,,C(1,6))                                    68.0311
000272          CALL INCOL(1H*,M,ID,  D    ,-KARDS(MM))                     67.0726
000300          IF(NOFILE.GT.0) RETURN
000302          IF(ID(1).EQ.8HOLD DATA) GO TO 40                            67.0726
000307          CALL XMIT(3*KARDS(MM),D,C(1,6))                             67.0726
000316          IF(M1.NE.0) GO TO 202                                       67.0726
000322          IF(LINE+7+KARDS(MM).GT.MAXLN) CALL HEAD(1)                  67.0726
000333          CALL OUTCOL(TITLE,M,X,X,0)                                  67.0726
000336     202  CALL OUTCOL(X,M,ID,C(1,6),-KARDS(MM))                      67.0707
000351          IF(M2.GT.1) GO TO 22                                        67.0203
000357          DO 21 N=1,3                                                 67.0203
000360          C(N,10)=C(N,7)                                              67.0203
000363          IF(KUNITS.EQ.0) C(N,6)=C(N,6)/SMF**2                        67.0203
000366     21   CONTINUE                                                    67.0203
000370          GO TO 30                                                    67.0203
000371       22 DO 24  N=1,3                                                66.0601
000373          IF(C(N,3).LE.0.) GO TO 24                                   66.0601
000375     23   C(N,11)=C(N,11)*GACC
000400          C(N,12)=C(N,12)/SMD
000402          C(N,13)=C(N,13)*SMD
000403          C(N,14)=C(N,14)*SMD**2
000405          C(N,16)=C(N,16)*T*OPI                                       66.0601
000407          IF (KUNITS.NE.0) GO TO 24
000410          C(N,6)=C(N,6)/SMF**2
000413     24   CONTINUE
000415          IF(M2.LE.4) GO TO 30                                        67.0707
000420          CALL IN1(NCARD,-1)                                          67.0707
000422          CALL INCOL(1H*,M,ID,C(1,17),-NCARD)                         67.0707
000434          IF(NOFILE.GT.0) RETURN
000436          CALL OUTCOL(X,M,ID,C(1,17),-NCARD)                          67.0707
           C
000450     30   IF (M3.EQ.0) RETURN                                             637
000452          CP(13)=0.
000456          READ(5,34) PMTR,QUIT                                        67.0707
000465          IF(CHEKFIL (5)) RETURN                                      68.0209
000471          AEQ=PMTR(1)                                                 67.0726
000473          IF(IFU.EQ.8HOLD DATA) GO TO 40                              67.0726
000477          IF(IEU.EQ.8HEND DATA) RETURN                                67.0726
000501          IF (LINE+15.GT.MAXLN) CALL HEAD(1)                              63M
000511          CALL OUTSET(120,12H           ,12H           ,12H           ,M)    639
000515          CALL OUTSET( 34,12H  MANEUVER  ,12H  AMPLITUDE ,12H  GRAVITIES ,M)    640
000521          FACTOR = GACC                                               67.0707
000523          IF(FLAG.EQ.BLANK) GO TO 31                                  67.0707
```

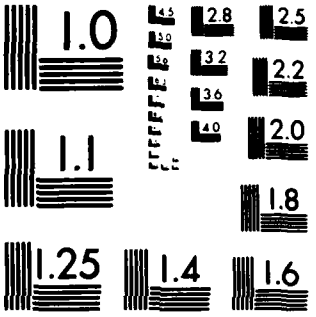
```
000527          FACTOR = 1./SRD                                                 67.0707
000531          CALL OUTSET(34,10H  ANGLE OF,9H   ATTACK,10H   DEGREES,4H)     67.0707
000535    31    CALL OUTSET( 44,12H  MANEUVER  ,12H  DIRECTION ,12H   DEGREES  ,4H)     641
000541          CALL OUTSET( 54,12H  DIRECTION ,12H     RATE    ,12H   DEG/SEC  ,4H)     642
000545          CALL OUTSET( 70,12H       CRITER,12H            ,12H PARAMETER  ,4H) 67.0707
000551          CALL OUTSET( 80,12HION FOR TERM,12H OF MANEUVER,12H UNITS      ,4H) 67.0707
000555          CALL OUTSET( 94,12HINATION     ,12H            ,12H    VALUE   ,4H)     645
000561          IF(M1.EQ.0 .AND. M2.EQ.0) CALL TITLER(-1,TITLE)                 67.0707
000577          CALL OUTCOL(1H ,M,A,X,0)                                             646
000603          CP(13)=1.                                                            649
000607          CP(14)=CP(15)=0.
000612          I1=16                                                           67.0707
000613          GO TO 35                                                        67.0707
000613    33    I1=I1+5                                                         66.1222
000615          READ (5,34) (PMTR(I),I=1,10),QUIT
000630    34    FORMAT(A4,A2,3F10.0,A4,A2,A8,A2,F10.0,A2)
000630          IF(CHEKFIL (5)) RETURN                                          68.0209
000634          AEQ=PMTR(1)                                                     67.0726
000636          IF(IEQ.EQ.8HEND DATA) RETURN                                    67.0726
000640    35    IEQ=1H                                                          69.0718
000642          CALL MCHAR(1,PMTR(8),1,IEQ,3)                                   69.071A
000646          CP(I1) = PMTR(3)*FACTOR                                         67.0707
000654          CP(I1+1)=PMTR(4)/SRD                                                 660
000656          CP(I1+2)=PMTR(5)/SRD                                                 661
000660          CP(I1+3)=PMTR(10)
000661          CP(I1+4)=PMTR(A)
000663          CALL OUTCOL(TITLE,M,X,PMTR,-1)                                       663
000666          DO 36 N=1,NNV                                                   69.0718
000672          IF(NAMVAL(1,N).NE.IEQ) GO TO 36                                 69.0718
000675          IEQ=NAMVAL(KUNITS*2,N)                                          69.0718
000700          CP(I1+3)=CP(I1+3)*AEQ                                           69.0718
000703          GO TO 37                                                        69.0718
000703    36    CONTINUE                                                        69.0718
000710    37    CP(I1+5)=CP(I1+9)=0.                                            69.0718
000712          IF (QUIT.NE.4HLANK) RETURN                                           664
000714          GO TO 33                                                        66.1222
        C
000715    40    CALL OLDATA(4HFLIN)                                             67.0726
000717          IF(M3.EQ.0) RETURN                                              67.0726
000721          CP(13)=1.  &  CP(14)=CP(15)=0.                                  67.0726
000730          RETURN                                                          67.0726
000730          END  FFLIN
```

```
      FUNCTION FOAL (STATE,GOAL)
C     SOURCE DATE 69.0709    CORRECT ARGUMENTS TO STALE
C     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.0818    REPAIR AZIM CALC FOR ROUND-EARTH
C     SOURCE DATE 67.0714   CALL TRAIDERR IF ERROR CONDITION
C     SOURCE DATE 67.0710    CALL ALTF FOR ALTITUDE
C     SOURCE DATE 66.1021     GOAL ETC ARE TYPE INTEGER
C     SOURCE DATE 66.0601   EXPAND TO ACCEPT 70 KINDS OF GOALS
C
C         RETURNS CURRENT VALUE OF FLIGHT GOAL
C
C         STATE  =  CURRENT STATE VECTOR IN FLIGHT
C         GOAL   =  HOLLERITH DEFINITION OF FLIGHT GOAL, IDENTICAL TO
C                   ONE OF THE POSSIBILITIES TABULATED BELOW
C
C         POSITION      COORDINATE    COORDINATE    RECT. VEL.    RECT. ACC.
C     ----COORD.--------1ST DERIV.---2ND DERIV.---COMPONENTS---COMPONENTS-----
C
C         X             X VEL.        X ACC.        X DOT         X DBLDOT
C         Y             Y VEL.        Y ACC.        Y DOT         Y DBLDOT
C         Z             Z VEL.        Z ACC.        Z DOT         Z DBLDOT
C         R             R DOT         R DBLDOT      R, VEL.       R, ACC.
C         THETA         THETA DOT     THETA DBLD    THETA VEL.    THETA ACC.
C         PHI           PHI DOT       PHI DBLDOT    PHI VEL.      PHI ACC.
C         RANGE         RANGE RATE    RANGE ACC.    RANGE VEL.    RANGE ACC.
C         AZIMUTH       AZ. RATE      AZ. DBLDOT    AZ. VEL.      AZ. ACC.
C         ELEVATION     EL. RATE      EL. DBLDOT    EL. VEL.      EL. ACC.
C         ALTITUDE      ALT. RATE     ALT. ACC.     VERT. VEL.    VERT. ACC.
C         LONGITUDE     LONG. RATE    LONG. ACC.    EAST VEL.     EAST ACC.
C         LATITUDE      LAT. RATE     LAT. ACC.     NORTH VEL.    NORTH ACC.
C
C         POLAR VEL.    POLAR ACC.    ACC. REL.     TIME
C     ----COORDINATE---COORDINATE---TO VEL.-------COORDINATE---
C
C         VEL. MAG.     ACC. MAG.     ACC. PAR.     TIME
C         VEL. AZ.      ACC. AZ.      ACC. NOR.     TIME INT.
C         VEL. EL.      ACC. EL.
C
      COMMON,BASCON
000005        COMMON /BASCON/
     1   PROGRM,    KPAGE,    LINE,     TOFDAY,   RUN,      RUNID(6),
     2   MSG,       FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
     3   KLASS,     KGROUP,   KUNITS,   KSIZNT,   KOUND,    IFEOF
     4  ,NOFILE
      COMMON,MASCON
C                                                                    09/15/69
      COMMON,CONCON
000005        COMMON /CONCON/
     1   PI,        SRD,      SLV,      SMF,      SRP,      RMOBY,
     2   GACC,      GCON,     RBODY,    RMOZRO,   TWOPI,    HAFPI
      COMMON,CONCON
C                                                                    09/15/69
000005        DIMENSION STATE(10),S(10),GOALS(70),GOLD(5),M1(5),M2(5),A(10),Y(3)
000005        DATA ((GOALS(I), I = 1,70)=
C
C              1         2         3         4         5                    68.0209
     *  5HTIME    X         Y         Z         X VEL.    ,                68.0209
```

AD-A127 695 THE COMPLEAT TRAIDSMAN(U) GENERAL RESEARCH CORP SANTA BARBARA CA T PLAMBECK 01 SEP 69 GRC-IM-711/2 2/3

UNCLASSIFIED F/G 9/2 NL





```
C M1,M2=    1, 1      1, 2      1, 3      1, 4      1, 5
C                                                              68.0209
C          6         7         8         9         10          68.0209
      *  50HY VEL.    Z VEL.    A ACC.    Y ACC.    Z ACC.    ,  68.0209
C M1,M2=    1, 6      1, 7      1, 8      1, 9      1,10
C                                                              68.0209
C          11        12        13        14        15          68.0209
      *  50HTIME INT  A DOT     Y DOT     Z DOT     A DBLDOT  ,  68.0209
C M1,M2=    1, 1      1, 2      1, 3      1, 4      1, 5
C                                                              68.0209
C          16        17        18        19        20          68.0209
      *  50HY DBLDOT  Z DBLDOT  M         RANGE     ALTITUDE  ,  68.0209
C M1,M2=    1, 6      1, 7      2, 2      2, 2      2, 2
C                                                              68.0209
C          21        22        23        24        25          68.0209
      *  50HVEL. MAG  ACC. MAG  THETA     AZIMUTH   LONGITUD  ,  68.0209
C M1,M2=    2, 5      2, 8      3, 2      3, 2      3, 2
C                                                              68.0209
C          26        27        28        29        30          68.0209
      *  50HVEL. AZ.  ACC. AZ.  PHI       ELEVATIO  LATITUDE  ,  68.0209
C M1,M2=    3, 5      3, 8      4, 2      4, 2      4, 2
C KOORD=2  11, 5     11, 8      4, 2      4, 2      4, 2
C                                                              68.0209
C          31        32        33        34        35          68.0209
      *  50HVEL. EL.  ACC. EL.  M DOT     RANGE RA  ALT. RAT  ,  68.0209
C M1,M2=    4, 5      4, 8      5, 2      5, 2      5, 2
C KOORD=2  12, 5     12, 8      5, 2      5, 2      5, 2
C                                                              68.0209
C          36        37        38        39        40          68.0209
      *  50HM VEL.    RANGE VE  VERT. VE  ACC. PAR  ACC. NOR  ,  68.0209
C M1,M2=    5, 2      5, 2      5, 2      5, 5      6, 0
C                                                              68.0209
C          41        42        43        44        45          68.0209
      *  50HTHETA DO  THETA VE  AZ. RATE  AZ. VEL.  LONG. RA  ,  68.0209
C M1,M2=    7, 0      7, 1      7, 0      7, 1      7, 0
C                                                              68.0209
C          46        47        48        49        50          68.0209
      *  50HEAST VEL  PHI DOT   PHI VEL.  EL. RATE  EL. VEL.  ,  68.0209
C M1,M2=    7, 1      8, 0      8, 1      8, 0      8, 1
C                                                              68.0209
C          51        52        53        54        55          68.0209
      *  50HLAT. RAT  NORTH VE  M DBLDOT  THETA DB  PHI DBLD  ,  68.0209
C M1,M2=    8, 0      8, 1      9, 8      9, 9      9,10
C                                                              68.0209
C          56        57        58        59        60          68.0209
      *  50HRANGE AC  AZ. DBLD  EL. DBLD  ALT. ACC  LONG. AC  ,  68.0209
C M1,M2=    9, 8      9, 9      9,10      9, 8      9, 9
C                                                              68.0209
C          61        62        63        64        65          68.0209
      *  50HLAT. ACC  M ACC.    THETA AC  PHI ACC.  RANGE AC  ,  68.0209
C M1,M2=    9,10     10, 8     10, 9     10,10     10, 8
C                                                              68.0209
C          66        67        68        69        70          68.0209
      *  50HAZ. ACC.  EL. ACC.  VERT. AC  EAST ACC  NORTH AC   )  68.0209
C M1,M2=   10, 9     10,10     10, 8     10, 9     10,10
C
000004        DATA (NOLLS=5),(NGOAL=0), ((GOLD(I), I=1,5)=5(-1)),(NOLD=1)   69.0709
000005        INTEGER GOAL,GOLD,GOALS                                       68.1021
```

```
      C
000005          DO 2  I=1,10
000008    2     S(I) = STATE(I)
000012          IF (GOAL.EQ.GOLD(NOLD)) GO TO 30                          69.0709
000015          DO 3  NOLD=1,NOLDS
000018          IF (GOAL.EQ.GOLD(NOLD)) GO TO 30
000020    3     CONTINUE
000023          DO 5 NGOAL = 1,70
000024          IF (GOAL.EQ.GOALS(NGOAL)) GO TO 10
000026    5     CONTINUE
000030          CALL READ(15)
000032          WRITE (6,8) GOAL,GOAL
000043    8     FORMAT(3(A56H* * *  GOAL DOES NOT RECOGNIZE THE FOLLOWING GOAL  *
                1* * /50X020/60AA10)                                    68.0209
000043          CALL STOUT(1H ,1111,A,5,1)
000047          CALL TRADEMR(1H )                                       68.0209
000051          RETURN                                                  67.0714
000053       10 DO 12 K=2,NOLDS
000056          NOLD=NOLDS-K+1
000060          GOLD(NOLD+1)=GOLD(NOLD)
000063          M1(NOLD+1)=M1(NOLD)
000065    12    M2(NOLD+1)=M2(NOLD)
000071       13 NOLD=1
000072          GOLD(1)=GOALS(NGOAL)
000074          IF(NGOAL.GT.17) GO TO 14
000100          M1(1) = 1
000100          M2(1) = NGOAL
000101          IF (NGOAL.GT.10)M2(1)=NGOAL-10
000105          GO TO 30
000106    14    IF (NGOAL.GT.32) GO TO 16
000112          M1(1)=(NGOAL-18)/5+2
000115          K=MOD(NGOAL-18,5)
000121          M2(1) = 2
000122          IF (K.LT.3) GO TO 30
000125          M2(1) = 2*3**(K-2)
000127          IF (NOOM(..NE.2.OR.M1(1).EQ.2) GO TO 30
000136          M1(1)=M1(1)*8                                           67.0818
000137          GO TO 30
000140    16    IF (NGOAL.GT.40) GO TO 18
000144          M1(1) = 5
000144          M2(1) = 2
000145          IF (NGOAL.LT.39) GO TO 30
000150          M2(1) = 5
000150          IF (NGOAL.LT.40) GO TO 30
000152          M1(1) = 6
000153          M2(1) = 0
000154          GO TO 30
000155    18    IF (NGOAL.GT.52) GO TO 20
000161          M1(1)=(NGOAL-41)/6+7
000164          M2(1)=MOD(NGOAL-41,2)
000171          GO TO 30
000171    20    M1(1)=(NGOAL-53)/9+9
000175          M2(1)=MOD(NGOAL-53,3)+8
000203    30    MODE1=M1(NOLD)
000205          MODE2=M2(NOLD)
000210    35    GO TO(41,46,51,56,61,66,71,76,81,81,91,96),MODE1
000230    41    FOAL=S(MODE2)
000233          RETURN
```

```
                      SUBROUTINE GRAV(S,G)                                          711
                C     SOURCE DATE 64.0728
                C                                                                   712
                C         RETURNS GRAVITATIONAL ACCELERATION G FOR STATE VECTOR S   713
                C                                                                   714
                C         WRITTEN 7/28/64                                           715
                C                                                                   716
                CCOMPKG,BASCON
000004                COMMON /BASCON/
                     1   PROGRM,    KPAGE,    LINE,     TOFDAY,   RUN,     RUNID(6),
                     2   MSG,       FLAG,     DATE,     MAXTM,    MAXPG,   MAXLN,
                     3   KLASS,     KGROUP,   KUNITS,   KSTINT,   KOORD,   IFEOF
                     4  ,NOFILE
                C
                CCOMPKG,CONCON
000004                COMMON /CONCON/
                     1   PI,        SMD,      SLV,      SMF,      SKP,     RBODY,
                     2   GACC,      GCON,     WBODY,    RMOZRO,   TWOPI,   HAFPI
                C
000004                DIMENSION S(10),G(3),P(3)                                   720
                C
000004                DO 5 I=2,4                                                  721
000006                P(I-1)=S(I)                                                 722
000010           5    G(I-1)=0.                                                   723
000014                IF(KOORD-1) 8,6,9
000016           8    G(3)=-GACC                                                  725
000021                RETURN                                                      726
000022           6    P(3)=P(3)+RBODY                                             727
000024           9    C=P(1)**2+P(2)**2+P(3)**2                                   728
000030                C=-GCON/(C*SQRT(C))                                         729
000035                DO 7 I=1,3                                                  730
000037           7    G(I)=C*P(I)                                                 731
000045                RETURN                                                      732
000045                END  GRAV
```

```
      SUBROUTINE HEAD(N)                                                734
C     SOURCE DATE 69.0709   FACILITATE DISCONNECT OF TIME/PAGE LIMITS
C     SOURCE DATE 68.1205   CORRECT WRITE(6,24) STATEMENT
C     SOURCE DATE 68.0403   ADD PROVISION FOR SUB-HEADING
C     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.0726   CALL OLDATA ON READING SPECIAL DATA-CARD
C     SOURCE DATE 67.0626   CHANGE COMPANY NAME
C     SOURCE DATE  67.0324   PRINT RUN-END UPON READING EOF
C     SOURCE DATE  66.1222    RE-ARRANGE EXIT ON CALL WITH (0)
C     SOURCE DATE 66.1020     NOFILE IS COUNT OF EOF-S FOUND
C                                                                       735
C         FOR N(1) NOT (MAGNITUDE GREATER THAN 99), TAKES N(1) AS       736
C                  PROGRAM NAME FOLLOWED BY (OPTIONAL) HOLLERITH TITLE, 737
C                  READS NEW HEADING CARD, WRITES NEW TITLE PAGE,       738
C                  RESETS PAGE COUNTER TO ZERO.                         739
C         FOR N(1) POSITIVE, SKIPS TO NEW PAGE WITH APPROPRIATE         740
C                  SECURITY MARKINGS, READS IT, AND SKIPS DOWN N(1)     741
C                  SPACES.                                              742
C         FOR N(1) ZERO, WRITES OUT RUN TERMINATION                     743
C                                                                       744
C         COLUMNS    FORMAT    CONTENTS                                745
C          -------    -------  --------------------------------        746
C          1-2        A2       RUN IDENTIFICATION NUMBER               747
C          3-50       6A8      RUN TITLE
C          51         A1       MESSAGE FLAG.  IF NOT BLANK,  CAUSES    749
C                              HEADING AND PRINTING OF MESSAGE CARDS   750
C                              (FORMAT 13A6,A2) UNTIL CARD CONTAIN-    751
C                              ING AN ASTERISK IN COL. 80 IS FOUND.    752
C          52         A1       INPUT FLAG, PLACED IN COMMON.           753
C          53-60      A8       RUN DATE (IF BLANK, TAKEN FROM SYSTEM
C                              DATE CELL)                              755
C          61-63      I3       EXECUTION TIME LIMIT IN MINUTES  (IF    756
C                              BLANK OR ZERO, RESET BY HEAD TO 5)      757
C          64-66      I3       OUTPUT PAGE LIMIT (IF BLANK OR ZERO,    758
C                              RESET BY HEAD TO 100)                   759
C          67-68      I2       NUMBER OF LINES TO BE WRITTEN ON OUT-   760
C                              PUT PAGES (IF BLANK OR ZERO, RESET BY   761
C                              HEAD TO 57)                             762
C          69         I1       CONTROLS SECURITY MARKINGS ON OUTPUT    763
C                              0 = UNCLASSIFIED                        764
C                              1 = CONFIDENTIAL                        765
C                              2 = SECRET                              766
C                              3 = TOP SECRET                          767
C                              4 = CONFIDENTIAL RESTRICTED DATA        768
C                              5 = SECRET RESTRICTED DATA              769
C                              6 = TOP SECRET RESTRICTED DATA          770
C          70         I1       SPECIFIES DOWNGRADING GROUP FOR         771
C                              SECURITY MARKINGS                       772
C          71         I1       CONTROLS UNITS IN BASIC COMPUTATIONS    773
C                              0 OR BLANK = METRIC (MKS) UNITS         774
C                              1 = ENGLISH (FEET, POUNDS, POUNDALS)    775
C          72         I1       CONTROLS INTEGRATION METHOD USED BY     776
C                              SUBROUTINE STINT                        777
C                              0 OR BLANK = RECTANGULAR                778
C                              1 = FOURTH/UNDER RUNGE-KUTTA            779
C                                                                      780
C                                                                      781
```

PRECEDING PAGE BLANK-NOT FILMED

```
                                                                                                                    7M2
      C          WRITTEN 7/6/64                                                                                     7M3
      C
      CCOMPKG,MASCON
000003       COMMON /BASCON/
           1   PROGRM,    KPAGE,     LINE,      TUFDAY,    RUN,       RUNID(6),
           2   MSG,       FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
           3   KLASS,     KGROUP,    KUNITS,    KSTINT,    KOORD,     IFEOF
           4   ,NOFILE
      CCOMPKG,MASCON                                                                                  09/15/69
      C
      CCOMPKG,CONCON
000003       COMMON /CONCON/
           1   PI,        SMD,       SLV,       SMF,       SKF,       MBODY,
           2   GACC,      GCON,      WBODY,     WBODMO,    TWOPI,     HALFPI
      CCOMPKG,CONCON                                                                                  09/15/69
      C                                                                                                   69.0709
      CCOMPKG,HEDSAV
000003       COMMON /HEDSAV/   TSTART,  IPG,  KUNTHL                                                    69.0709
      CCOMPKG,HEDSAV                                                                                  09/15/69
      C
000003       DIMENSION N(2),CLASS(6,6),GROUP(8,4),CUNDX(2,4),CONTRL(2),                                 68.0209
     *           WORDS(15),KTIME(3),IMAGE(8)                                                          68.0209
000003       DATA  (CLASS =
     1 32H--------  CONFIDENTIAL  -------- ,                                                              790
     2 32H----------  SECRET  ---------- ,                                                              791
     3 32H----------  TOP SECRET  ---------- ,                                                         792
     4 32H  CONFIDENTIAL RESTRICTED DATA   ,                                                          793
     5 32H---  SECRET RESTRICTED DATA  --- ,                                                          794
     6 32H-  TOP SECRET RESTRICTED DATA  - )                                                          795
000003       DATA  ( GROUP =
     1 72HT        EXCLUDED FROM AUTOMATIC DOWNGRADING AND DECLASSIFICATI                             797
     2ON         1 ,                                                                                  798
     3 72HT        EXCLUDED FROM AUTOMATIC DOWNGRADING AND DECLASSIFICATI                             799
     4ON         1 ,                                                                                  800
     5 72HT   DOWNGRADED AT 12-YEAR INTERVALS,  NOT AUTOMATICALLY DECLAS                              801
     6SIFIED     1 ,                                                                                  802
     7 72HT     DOWNGRADED AT 3-YEAR INTERVALS,  DECLASSIFIED AFTER 12 Y                              803
     8EARS       1 )                                                                                  804
000003       DATA (CONTRL=1H ,1H1),(ASTRSK=1H*),(DOT=1H.)                                               68.0209
000003       EQUIVALENCE (ABC,IBC),(BLANK,CONTRL)
000003       DATA (CONDX=12H       METRIC,  12H     ENGLISH,
     *             12H RECTANGULAR,  12H RUNGE-KUTTA)

      C
000003       TUFDAY=UTIMEF(X)                                                                           809
000006       IF (IABS(N(1)).LT.99) GO TO 50                                                             810
000012       READ(5,99) IMAGE                                                                         67.0726
000017    99 FORMAT(8A10)                                                                             68.0209
000017       IF(EOF,5) 79,3                                                                           67.0726
000023    3  IF(IMAGE(1).NE.8HOLD DATA) GO TO 5                                                       67.0726
000026       CALL OLDATA(4HHEAD)                                                                      67.0726
000027       RETURN                                                                                   67.0726
000030    5  DECODE(80,2,IMAGE) RUN,(RUNID(I),I=1,5),                                                68.0209
     *                                  MSG,FLAG,DATE,MAXTM,MAXPG,MAXLN,
                                                                                                        813
     1 KLASS,KGROUP,KUNITS,KSTINT                                                                    68.0209
000073    2  FORMAT(A2,4A10,AH,
     *              2A1,A6,2I3,I2,4I1)
000073       RUNID(6)=BLANK                                                                           68.0209
000075       IF (MAXTM.EQ.0) MAXTM=999                                                                69.0709
```

```
000077    1     CALL SECOND(TSTART)                                                    68.0209
000101          IF (MAXPG.EQ.0) MAXPG=999                                              69.0709
000103          IF (MAXLN.EQ.0) MAXLN=57                                                  81M
000106          IF (KLASS.NE.0) MAXLN=MAXLN-4                                             819
000110          IF (DATE.EQ.6HBLANK) DATE=DATEF(X)                                        820
000116          DUMMY=RANF(*3.141592654)                                               68.0209
000121          KOUNT=0                                                                   822
000122          IPG=0                                                                     823
000123    4     IF (KLASS.EQ.0) WRITE (6,6)                                               824
000131    6     FORMAT(1H1)                                                               825
000131          IF (KLASS.NE.0) WRITE (6,7) (CLASS(I,KLASS),I=1,4)                        826
000147    7     FORMAT(1H1 43(1H-) 3A10,A2, 44(1H-) / 1H0)                             68.0209
000147    10    LINE=1                                                                    828
000150          CALL LSKIP((MAXLN-41)/2)                                                  829
000154          CALL TITLER(0,N)                                                          830
000160          WRITE (6,11) RUN,DATE,TUEDAY                                              831
000172    11    FORMAT(1H0 41X4HRUN A2,5XA10,4X5HTIME A10)                             68.0209
000172          CALL TITLER(0,RUNID)                                                      833
000174    16    WRITE (6,17) (CUNDX(I,KUNITS+1),I=1,2),(CUNDX(I,KSTINT+3),I=1,2),      834
              1 MAXTM,MAXPG                                                             835
000222    17    FORMAT(1H0/1H  39X 17HCOMPUTATION UNITS      6A1H= 4X A10,A2          68.0209
              1              /40X18HINTEGRATION METHOD      5A1H= 4X A10,A2          68.0209
              2              /40X18HRUNNING TIME LIMIT      5A1H= I8, 8H MINUTES        83M
              3              /40A12HOUTPUT LIMIT            11A1H= I10, 6H PAGES        839
              4 /1H1/1H0 )                                                              840
000222          LINE=LINE+12                                                              841
000224    20    IF (KLASS.EQ.0) GO TO 29                                                  842
000226          WRITE (6,21)                                                              843
000232    21    FORMAT(24X72(1H-)/24X72HI THIS MATERIAL CONTAINS INFORMATION AFFEC        844
              1TING THE NATIONAL DEFENSE OF I /24X72HI THE UNITED STATES, WITHIN        845
              2 THE MEANING OF THE ESPIONAGE LAWS, TITLE I  /24X72HI 18, U.S.C.,        846
              3  SECTION 793 AND 794,  THE TRANSMISSION OR REVELATION OF I  /           847
              4 24X72HI WHICH IN ANY MANNER TO AN UNAUTHORIZED PERSON IS PROHIBI        848
              5ED BY LAW.  I )                                                          849
000232          LINE=LINE+5                                                               850
000234          IF (KLASS.GE.4) KGROUP=1                                                  851
000237          WRITE (6,24) KGROUP,(GROUP(I,KGROUP),I=1,8)                            68.1205
000255    24    FORMAT(24X1HI 70A1HI / 24A 1HI 31X 5HGROUP I3,31X 1HI /24X7A10,A2)     68.0209
000255          LINE=LINE+3                                                               854
000257          IF (KLASS.LT.4) GO TO 26                                                  855
000262          WRITE (6,25)                                                              856
000266    25    FORMAT(24X1HI 70X 1HI / 24X 1HI 27X 15HRESTRICTED DATA 28X 1HI /          857
              1 24X 1HI 22X 25HATOMIC ENERGY ACT OF 1954 23X 1HI )                      858
000266          LINE=LINE+2                                                               859
000270    26    WRITE (6,27)                                                              860
000274    27    FORMAT(24X 72(1H-))                                                       861
000274          LINE=LINE+1                                                               862
000276    29    CALL LSKIP(9-2*(KLASS/4))                                                 863
000303    30    WRITE (6,34)                                                              864
000307    34    FORMAT(1H0/1H0 47X 24H       TRAID SYSTEM         /48X24H GENERAL RE 67.0626
              1SEARCH CORP.  /48X24H 5383 HOLLISTER AVENUE  /48X24H   GOLETA, CAL 69.0709
              2IFORNIA    / )
000307          LINE=LINE+8                                                               868
000311          IPG=IPG+1                                                                 869
000312          KOUNTL=2                                                                  870
000313          IF (KLASS.NE.0) KOUNTL=1                                                  871
000315          IF (IPG.EQ.2) GO TO 40                                                    872
000320          IF (KLASS.EQ.0) GO TO 4                                                   873
```

```
000320          GO TO 50                                                        874
000321     60   ITO=1
000322          PROGRM=BLU
000324          KPAGE=0                                                         876
000325          PI=3.1415v2654                                                  877
000326          TWOPI=2.*PI                                                   66.0601
000330          HALFPI=.5*PI                                                  66.0601
000331          SHF=57.29577951                                                 878
000333          SMF=3.280833                                                    879
000334          SKF=2.2046223                                                   880
000336          RBODY=.000072722052                                           69.0709
000337          SLV=1000.                                                       882
000341          RBODY=6371180.                                                  883
000342          GACC=9.80665                                                    884
000344          GCONV=GACC*RBODY**2                                             885
000346          RHO2HO=1.225                                                    886
000347          IF (KUNITS.EQ.0) RETURN                                         887
000351          SLV=ACHO.3                                                      888
000354          RBODY=RBODY*SMF                                                 889
000356          GACC=GACC*SMF                                                   890
000355          GCONV=GACC*RBODY**2                                             891
000357          RHO2HO=RHO2HO*SKF/SMF**3                                        892
000361          RETURN                                                          893
000362     50   CALL SECOND(TNOW)                                             68.0209
000364          TNOW=TNOW-TSTART                                              68.0209
000366          KTIME(1)=FLIV(TNOW/3600.)                                     68.0209
000372          KTIME(2)=AMOD(FDIV(TNOW/60.),60.) * 1000                      68.0209
000404          KTIME(3)=AMOD(TNOW*60.) * 1000
000411          IF (KLASS.EQ.0) GO TO 55                                        899
000413          CALL LSKIP(MAXLN-LINE)                                          900
000417          WRITE (6,54) ((CLASS(I,KLASS),I=1,6),K=1,2)                     901
000436     54   FORMAT(1HO 43(1H=) 3A10,A2,                                   68.0209
               *                          44(1H=) /1H1 43(1H=) 3A10,A2,      68.0209
               *                                                44(1H=) /1H )
000436          IF (IPG.EQ.1) GO TO 10                                          903
000441     55   IF(IPG.LE.-98) GO TO 791                                      67.0324
000444          KPAGE=KPAGE+1                                                 66.1222
000445          WRITE (6,56) CONTRL(KONTRL),PROGRM,RUN,(RUNID(I),I=1,5),DATE,   68.0209
               *  KTIME(1),DOT,KTIME(2),DOT,KTIME(3),KPAGE                     68.0209
000504     56   FORMAT(A1,A10,5H RUN A2,)1X,A10,AB,11XA10,2X,4HTIME            68.0209
               *    I2,2(1H:,I2)                                              68.0209
               1 7H   PAGE I4)                                                  908
000506          LINE=1                                                          909
000505          CALL SUBHEAD(6)                                               68.0603
000507          IF(IPG.EQ.-97) GO TO 90                                       67.0324
000512          IF (C(1).EQ.0) GO TO 89                                       67.0324
000513          IF(MSG.EQ.1H ) GO TO 76
000515          CALL LSKIP(10)                                                  913
000516     61   READ(5,62) WORLS                                                914
000524     62   FORMAT(13A6,2A1)                                                915
000524          LINE=LINE+1                                                     916
000526          IF (LINE.GT.MAXLN) GO TO 50                                     917
000532          IF (WORDS(15).EQ.ASTWSK) GO TO 65                               918
000534          WRITE (6,64) WORDS                                              919
000541     64   FORMAT(20X,13A6,2A1)                                            920
000541          GO TO 61                                                        921
000543     65   WRITE (6,64) (WORDS(I),I=1,14)                                  922
000555          MSG=1H
```

```
000557          GO TO 50                                                        924
000560       70 IF(MAXTM.GE.999) GO TO 71                                    69.0709
000563          IF (KTIME(2)-1000*60*KTIME(1).GE.MAXTM) GO TO 75             69.0709
000570       71 IF(MAXPG.GE.999) GO TO 72                                    69.0709
000573          IF (KPAGE.GT.MAXPG) GO TO 77                                    926
000576       72 CALL LSKIP(A(1))                                             69.0709
000600          RETURN                                                          928
000601       75 CALL LSKIP(20)                                                  929
000603          WRITE (6,76)                                                    930
000607       76 FORMAT(27X45H*   *   *   JOB TERMINATED FOR EXCESSIVE RUNNING TIME   931
              1   *   *   *   )                                                 932
000607          GO TO 80                                                        933
000611       77 CALL LSKIP(20)                                                  934
000613          WRITE (6,78)                                                    935
000617       78 FORMAT(33X54H*   *   *   JOB TERMINATED AT OUTPUT LIMIT   *   *   936
              1*   )                                                           937
000417          GO TO 80                                                     66.0601
000621       79 NOFILE=NOFILE+1                                              66.1020
000623          IF(IFEOF.GT.0) RETURN                                        68.1205
000625          IPG=-97                                                      67.0324
000628          GO TO 50                                                     66.0601
000627       791 IF(IPG.EQ.-99) RETURN                                       67.0324
000632       80 CALL EXIT                                                    66.0601
000633       89 IPG=-98                                                      67.0324
000634       90 IPG=IPG-1                                                    67.0324
000638       91 CALL LSKIP(15)                                               66.0601
000640          CALL TITLER(0,16H=  END OF RUN  =)                               940
000642          WRITE (6,92) KTIME                                               941
000650       92 FORMAT(1H0/1H0/1H0 42X 15HEXECUTION TIME I2, 6H HRS  I2, 6H MIN   942
              1 I2, 4H SEC  )                                                   943
000650          LINE=LINE+6                                                      944
000652          GO TO 50                                                         946
000653          END  HEAD
```

```
      SUBROUTINE IN1(DATA,INP)                                          68.0209
C     SOURCE DATE 69.0709    ADD 2 UNIT-NAMES, REVISE FORMAT IF NAME USED
C     SOURCE DATE 69.0220   CORRECT OUTPUT FORMAT
C     SOURCE DATE 68.1205  PRINT ASTERISK IF DATA-VALUE CONVERTED TO INTEGER
C     SOURCE DATE  68.0527   MAKE CONVERSIONS DEPEND ON KUNITS
C     SOURCE DATE 68.0507   ADD BCD READ IF INP = ZERO
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.0726   CALL OLDATA ON READING SPECIAL DATA-CARD
C     SOURCE DATE 67.0711   ALLOW UNIT-NAMES FOR AUTOMATIC CONVERSION
C     SOURCE DATE 63.1101
C                                                                          949
C         READS AN IDENTIFICATION, A NUMBER, AND TWO SCALINGS FROM EACH    950
C         OF  -INP-   CARDS OF THE FOLLOWING FORMAT -
C                                                                          952
C              COLUMN     FORMAT     CONTENTS                            953
C              ------     ------     ------------------------------------ 954
C                                                                          955
C              1-40       5A8        HOLLERITH IDENTIFICATION
C              41-50      F10.0      AN INPUT NUMBER, X                   957
C              51-60      F10.0      A MULTIPLIER, XM OR UNIT-NAME (BELOW)
C              61-70      F10.0      A DIVISOR, XD                        959
C                       ALTERNATIVELY, COL 51-53 MAY CONTAIN A UNIT-NAME
C                       FOR AUTOMATIC CONVERSION, THEN COL 54-80 ARE IGNORED
C                       UNIT-NAME PUNCHED  INTERNAL(MET.)   (ENGL.)
C                                 -------  ----------       ---------
C                                 DEG      RADIANS          RADIANS
C                                 FT       METERS           FEET
C                                 PSF      KGM/SQ METER     LM/SQ FT
C                                 KM       METERS           FEET
C                                 NMI      METERS           FEET
C                                 LB       KGM              LB
C                                 G        METER/SEC SQ     FT/SEC SQ
C                                 KFT      METERS           FEET
C                                 M        METERS           FEET
C                                 MRA      RADIANS          RADIANS      69.0709
C                                 SEC      SECONDS          SECONDS      69.0709
C                                                                          960
C         RETURNS DATA(I)=X*XM/XD, WHERE X, XM, AND XD ARE TAKEN FROM      961
C         THE I-TH INPUT CARD.  IF BLANK, XM AND XD ARE TAKEN AS UNITY.    962
C                                                                          963
C         PRINTS IDENTIFICATION AND X, WITH SCALED VALUE X*XM/XD IF IT     964
C         DIFFERS FROM X                                                   965
C                                                                          966
C         FOR  -INP-   POSITIVE, FLOATING POINT VALUES ARE RETURNED
C         FOR  -INP-   NEGATIVE, INTEGER VALUES ARE RETURNED
C         FOR  -INP-   ZERO, BCD DATA IS RETURNED
C                                                                          969
C                                                                          971
      CCOMPKG,HASCON
000005        COMMON /HASCON/
     1    PROGRM,    KPAGE,     LINE,      TODAY,     RUN,       RUNID(6),
     2    MSG,       FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
     3    KLASS,     KGROUP,    KUNITS,    KSTINT,    KOORD,     IFEOF
     4   ,NDFILE
      CCOMPKG,HASCON
C                                                                  09/15/69
      CCOMPKG,COMVRT                                                69.0709
```

```
000005          COMMON /CONVRT/ NNV, NAMVAL(3,11)                          69.0709
000005          DATA (NNV=11)                                              69.0709
000005          DATA (NAMVAL =                                             68.0527
               1   3HDEG,  2(.01745329252),                              68.0527
               2   2HFT,   0.3048H006,  1.0,                             68.0527
               3   3HPSF,  4.88024,  1.0,                                68.0527
               4   2HNM,   1000.,  3280.833,                             68.0527
               5   3HNMI,  1853.25,  6080.3,                             68.0527
               6   2HLB,   0.453592,  1.0,                               68.0527
               7   1HG,    9.80665,  32.17398,                           68.0527
               8   3HMFT,  304.H006,  1000.,                             68.0527
               9   1HM,    1.0,   3.280833 ,                             69.0709
               0   3HMRA,  2(.001) ,                                     69.0709
               1   3HSEC,  2(1.)  )                                      69.0709
         CCOMPKG,CONVRT                                                   69.0709
         C                                                              09/15/69
         C
000005          DIMENSION IMAGE(7),DATA(INP)
000005          EQUIVALENCE (AEQ,IEQ)
000005          EQUIVALENCE (MA,XM)                                        68.0527
000005          LOGICAL NAMED                                              69.0709
         C
000005          KAR=1H    $  IF(INP.LT.0) KAR=1H*                          68.1205
000011          INPTS=IABS(INP)                                            68.0209
000013          DO 22 INPUT=1,INPTS                                           976
000014          READ(5,1) IMAGE
000021        1 FORMAT(7A10)                                               68.0209
000021          IF(CHEKFIL (5)) RETURN                                     68.0209
000025          IF(IMAGE(1).NE.8HOLD DATA) GO TO 6                         67.0726
000031          CALL OLDATA(3HIN1)                                         67.0726
000032          RETURN                                                     67.0726
000033        6 IF(INP.EQ.0) GO TO 30                                      68.0507
000035          DECODE(60,5,IMAGE) NAM,NAM2                                69.0709
000047        5 FORMAT(50X,A3,A7)                                          69.0709
000047          DO 2 K=1,NNV                                               68.0527
000052          IF(NAM.EQ.NAMVAL(1,K)) GO TO 4                             68.0527
000055    2     CONTINUE                                                   67.0711
000057          DECODE(70,3,IMAGE    ) X,XM,XD
000073          NAM=3H
000075          NAM2=1H                                                    69.0709
000076          NAMED=.FALSE.                                              69.0709
000077          GO TO 10                                                   67.0711
         C                                                                 67.0711
000101        4 MA=NAMVAL(KUNITS+2,K)
000105          NAMED=.TRUE.                                               69.0709
000106          XD=0.                                                      67.0711
000107          DECODE(50,3,IMAGE) X
000116        3 FORMAT(40X,3F10.0)                                         67.0711
000116   10     A=X                                                        67.0711
         C                                                                 67.0711
000120          IF (XM.NE.0.) A=A*XM                                          980
000122          IF (XD.NE.0.) A=A/XD                                          981
000125          IF (LINE+1.GE.MAXLN) CALL HEAD(4)                             982
000134          LINE=LINE+1                                                   983
000136          IF (X.NE.A  .OR.  NAMED) GO TO 19                          69.0709
000143          WRITE(6,1H) (IMAGE(K),K=1,4),X,KAR                         68.1205
000162          GO TO 21                                                      987
000164       18 FORMAT(12X,A10,F14.4,A1,                                   68.1205

                .                    A3,A7,7X  1H(G14.7,15H AFTER SCALING) )    69.0709
000164    19    WRITE(6,18) (IMAGE(K),K=1,4),X,KAR,NAM,NAM2,A              69.0709
000211    21    IF(INP    .GT.0) GO TO 22                                  68.0209
000215          IEQ=A
000216          A=AEQ
000217    22    DATA(INPUT)=A                                                 991
000224          RETURN                                                        992
         C
000224       30 IEQ=IMAGE(5)  $  A=AEQ                                     68.0507
000227          CALL COUNOUT(1)                                            69.0709
000231          WRITE(6,31) IMAGE                                          68.0507
000237       31 FORMAT(12X,A10,4X,3A10)                                    69.0220
000237          GO TO 22                                                   68.0507
         C
000241          END  IN1
```

```
      SUBROUTINE IN3(DATA,NIN)
C     SOURCE DATE 68.1205  PRINT ASTERISK IF DATA-VALUE CONVERTED TO INTEGER
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.0726   CALL OLDATA ON READING SPECIAL DATA-CARD
C     SOURCE DATE 67.0711   BRAND NEW CODE
C
C     READS NIN VALUES FROM IN1-TYPE CARDS, AND STORES THEM IN THE DATA
C     ARRAY. THE DIFFERENCE BEING THAT IN3 READS 3 VALUES FROM
C     COL 41-70 OF EACH CARD.
C
C
000005      DIMENSION DATA(NIN),ID(4),VAL(3)
000005      EQUIVALENCE(AEQ,IEQ)
C
C
000005      KAR=1H   $  IF(NIN.LT.0) KAR=1H*                          68.1205
000011      MAX = IABS(NIN)
000013      MX = (MAX+2)/3
000017      M = 0
000020      DO 6 MXX=1,MX
000021      READ(5,1) ID,VAL
000030    1 FORMAT(4A10,3F10.0)                                       68.0209
000030      IF(CHEKFIL (5)) RETURN                                    68.0209
000034      IF(ID(1).NE.6HOLD DATA) GO TO 4                           67.0726
000040      CALL OLDATA(3HIN3)                                        67.0726
000041      RETURN                                                    67.0726
000042    4 CALL COUN OUT(1)                                          68.0209
000044      WRITE(6,2) ID,VAL(1),KAR                                  68.1205
000060    2 FORMAT(12X,4A10,F14.4,A1)                                 68.1205
000060      IF(M+1.GE.MAX) GO TO 5
000065      CALL COUN OUT(1)                                          68.0209
000066      WRITE(6,3) VAL(2),KAR                                     68.1205
000100    3 FORMAT(F66.4,A1)                                          68.1205
000100      IF(M+2.GE.MAX) GO TO 5
000105      CALL COUN OUT(1)                                          68.0209
000106      WRITE(6,3) VAL(3),KAR                                     68.1205
000120    5 DO 6 K=1,3
000123      M = M+1
000125      IF(M.GT.MAX) RETURN
000130      IF(NIN.GT.0) GO TO 6
000133      IEQ=VAL(K) $ VAL(K)=AEQ
000140    6 DATA(M) = VAL(K)
C
000150      RETURN
000151      END  IN3
```

```
      SUBROUTINE INCOL(ITITLE,SETUP,NAMES,DATA,LINES)                    1131
C     SOURCE DATE 69.0709    WIDEN F.3 AND F.6 FORMAT-FIELDS
C     SOURCE DATE 68.0507    PROVIDE FOR OPEN-END READING
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.1027    REPAIR CHANGES OF 67.0726
C     SOURCE DATE 67.0726    CALL OLDATA ON READING SPECIAL DATA-CARD
C     SOURCE DATE  67.0406     CORRECTED FMTS ARRAY, KOLUMS COUNTER
C     SOURCE DATE 65.0106
C                                                                        1132
C         READS AND PRINTS COLUMNS OF DATA FROM CARDS                    1133
C                                                                        1134
C         ARGUMENTS ARE DEFINED IN SUBROUTINES OUTCOL AND OUTSET         1135
C                                                                        1136
C         OUTCOL PRINTS LINES OF 1 TO 10 NUMBERS IN COLUMNS, WITH        1137
C         HEADINGS AND TITLES.  INCOL USES OUTCOL FOR THIS PURPOSE       1138
C         AFTER FIRST READING THE LINES FROM CARDS.  IN DOING THIS,      1139
C         IT CONSIDERS CARDS TO BE DIVIDED INTO 7 10-CHARACTER COLUMNS,  1140
C         SO IF A LINE CONTAINS 8 OR MORE COLUMNS TWO CARDS ARE REQUIRED 1141
C         PER LINE.  INCOL DETERMINES FORMATS FOR EACH CARD COLUMN FROM  1142
C         EXAMINATION OF THE FORMATS IN SETUP TO BE USED BY OUTCOL IN    1143
C         PRINTING THE INPUT DATA.                                       1144
C                                                                        1145
C         INPUT DATA IN A LINE IS LOCATED ON CARD(S) AS FOLLOWS -        1146
C                                                                        1147
C         CARD 1  -  COLUMNS 1-10   -  WORD 1                            1148
C                            11-20  -  WORD 2                            1149
C                             AND SO ON FOR WORDS 3 - 7                  1150
C         CARD 2  -  COLUMNS 1-40   -  (NOT READ)                        1151
C                            41-50  -  WORD 8                            1152
C                            51-60  -  WORD 9                            1153
C                            61-70  -  WORD 10                           1154
C                                                                        1155
C         IF AN N-CHARACTER NAME IS REQUIRED FOR THE LINE, ITS N CHAR-
C         ACTERS APPEAR IN COLS. 1-N OF CARD 1 AND THE WORDS ARE MOVED N
C         CARD COLS. TO THE RIGHT IN CONSEQUENCE.
C                                                                        1159
C         FORMATS FOR DATA COLUMNS IN EACH CARD ARE DETERMINED FROM      1160
C         FORMATS IN SETUP OR THE CODES SELECTING THEM IN OUTSET         1161
C         AS FOLLOWS -                                                   1162
C                                                                        1163
C         CODE      OUTPUT FORMAT       INPUT FORMAT                   1164
C         ----      -------------       ------------                   1165
C         1         E12.4               E10.0
C         2         E11.2               E10.0
C         3         F11.6               F10.0                    69.0709
C         4         F11.3               F10.0                    69.0709
C         5         F10.0               F10.0
C         6         I8                  F10.0
C         7         I10                 F10.0
C         8         O12                 O10                          1173
C         9         A10                 A10                      68.0209
C         *         A8,A2               A8,A2
C                                                                        1176
C         DECIMAL POINTS PUNCHED IN THE CARDS OVERRIDE, OF COURSE,       1177
C         THOSE IN THE FORMATS.  IF THE OUTPUT FORMAT IS INTEGER, THE    1178
C         INPUT FLOATING POINT VALUE IS FIXED ACCORDINGLY.               1179
C                                                                        1180
C         IF THE FIRST WORD OF TITLE IS AN ASTERISK FOLLOWED BY BLANKS,  1181
C         OUTPUT IS OMITTED.                                             1182
C                                                                        1183
C         WRITTEN 1/8/65                                                 1184
C                                                                        1185
      CCOMPKG,FMMATS                                                  69.0709
000010        COMMON /FMMATS/  PFORMS(2,13)
000010        DATA (PFORMS =
     1 12HG12.5,          ,12HG11.4,1X       ,12HF11.6,1X       ,        69.0709
     2 12HF11.3,1X        ,12HF10.0,2X       ,12HI8,4X          ,        69.0709
     3 12HI10,2X          ,12HO12,           ,12HXA10,X         ,        68.0209
     4 12H1XA8,A2,1X      ,12H2X2A8,A4,2X    ,12H3X3A8,A6,3X    ,
     5 12H4X5A8,4X        )
      CCOMPKG,FMMATS                                                  69.0709
C                                                                  09/15/69
```

```
000010          DIMENSION FMTS(16),CARDFM(15),SETUP(82),NAMES(5,2),DATA(2),
                *          INTGR(10),IMAGE(8),IMAG2(8)                          68.0209
000010          DATA ((FMTS(I),I=1,16)= 2(6HE10.0,),5(6HF10.0,),6H010,  ,
                1 6HA11,  ,6HA4,,A2,,6H2A8,A4,  ,8H3A8,A6,  ,8H5A8,     ,6H/,6HX  ,      68.0209
                2 1H(,1H))
000010          DATA (BLANK=1H ),(ASTRSK=1H*)
000010          EQUIVALENCE (AEQ,IEQ)

000010          LSAV=LINES                                                       68.0507
000011          IF(LINES.EQ.0) GO TO 19                                          68.0209
000011          CALL AMIT(-10,0,INTGR)                                           67.0406
000014          CARDFM(1)=FMTS(15)
000016          NAMED=KOLUMS=0                                                   67.1027
000020          KHDCOL=NKARD=1                                                   67.1027
000022          DO 8 KCOL=1,10                                                      1200
000024          DO 2 MPRINT=1,13
000027          IF (SETUP(2*KCOL).EQ.MFORMS(1,MPRINT)) GO TO 4
000034    2     CONTINUE                                                            1203
000035          GO TO 8                                                             1204
000036    4     KHDCOL=KHDCOL+1
000040          IF (MPRINT.LT.11) GO TO 6
000042          NAMED=MPRINT-8
000044          CARDFM(KHDCOL)=FMTS(MPRINT)
000047          DO 5  I=11,MPRINT
000050          KHLCOL=KHLCOL+1
000052    5     CARDFM(KHDCOL)=BLANK
000056          GO TO 8                                                             1216
000057    6     KOLUMS=KOLUMS+1 + MPRINT/10                                      67.0406
000064          IF (MPRINT.EQ.6.OR.MPRINT.EQ.7) INTGR(KOLUMS)=1                     1208
000075          IF (KHDCOL.NE.9) GO TO 7
000077          CARDFM(KHDCOL)=FMTS(16)
000101          KHDCOL=KHLCOL+1                                                     1219
000102          NKARD=2                                                          67.1027
000103    7     CARDFM(KHDCOL)=FMTS(MPRINT)                                         1220
000107    8     CONTINUE                                                            1221
000111          CARDFM(KHDCOL+1)=FMTS(16)
000113    10    LYNES=IABS(LINES)                                                   1223
000115          DO 18 LYNE=1,LYNES                                                  1224
000116          READ(5,1) IMAGE                                                  67.0726
000123        1 FORMAT(8A10)                                                     68.0209
000123          IF(CHERFIL (5)) RETURN                                           68.0209
000127          IF(IMAGE(1).EQ.8HEND DATA) GO TO 20
000134          IF(IMAGE(1).NE.8HOLD DATA) GO TO 11                              67.0726
000136          NAMES(1,1)=IMAGE(1)                                              67.0726
000137          RETURN                                                           67.0726
000137       11 IF(NKARD.EQ.1) GO TO 13
000141          READ(5,1) IMAG2                                                  67.1027
000147          IF(CHERFIL (5)) RETURN                                           68.0209
000153       13 I2=LYNE*KOLUMS
000154          I1=I2-KOLUMS+1                                                      1226
000160          IF (NAMED.NE.0) GO TO 12                                            1227
000164          DECODE(80,CARDFM,IMAGE) (DATA(I),I=I1,I2)                        67.0726
000204          GO TO 14                                                            1229
000210    12    DECODE(80,CARDFM,IMAGE) (NAMES(J,LYNE),J=1,NAMED),
                *                        (DATA(I),I=I1,I2)                       67.0726
000244    14    KOLUM=0                                                             1231
000245          DO 16 I=I1,I2                                                       1232
000252          KOLUM=KOLUM+1                                                       1233
000254          IF(INTGR(KOLUM).EQ.0) GO TO 16
000256          IEQ=DATA(I)
000260          DATA(I)=AEQ
000262    16    CONTINUE                                                            1235
000265    18    CONTINUE                                                            1236
000267       19 IF (TITLE.NE.ASTRSK) CALL OUTCOL(TITLE,SETUP,NAMES,DATA,LSAV)    68.0507
000276          RETURN                                                              1238
000277       20 LSAV=ISIGN(LYNE-1,LINES)                                         68.0507
000303          LINES=LYNE-1                                                     68.0507
000303          IF(LSAV.NE.0) GO TO 19                                           68.0507
000304          RETURN                                                           68.0507
000305          END  INCOL
```

```
      SUBROUTINE INDEC (ARG,MM)
C     SOURCE DATE 69.0709   REMOVE ENCODE STATEMENTS
C     SOURCE DATE 69.0131   ALLOW PRINTOUT TO BE SUPPRESSED  BY CALL INLOU(0)
C     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 66.0801
C
C     FORMAT-FREE INPUT ROUTINE
C
C     ENTRIES     INDEC   RETURNS FLOATING-POINT NUMBERS
C                 ININT   RETURNS FIXED-POINT NUMBERS
C                             READS DATA CARDS UNTIL MM NUMBERS ARE FOUND,
C                             WHICH ARE STORED IN ARG.
C                             SCANS DATA CARDS UNTIL A LEGAL NUMBER-FIELD
C                             BEGINS, THEN CONTINUES SCANNING UNTIL
C                             THAT NUMBER-FIELD TERMINATES ON AN ILLEGAL
C                             CHARACTER OR A BLANK - AND REPEATS MM TIMES
C
C                             AN INTEGER FIELD FOLLOWED IMMEDIATELY BY
C                             A LEFT PARENTHESIS IS INTERPRETED AS A
C                             REPETITION FACTOR (SEE SAMPLE DATA-CARD)
C
C                           ... SAMPLE DATA CARD ...
C                               X,Y,Z COORD=2.34E5 U(REL TO RADAR),6000, Z(100)
C                           ... NUMBERS     ******  *                *****   ***
C
C                           ... WRONG DATA CARD ...
C                               X,Y,Z COORD=2.34 E5 U(REL TO RAD 2) 6000(100.100
C                           ... PROBABLE ERRORS *                   *     *   *
C
C     ENTRY       INALF   RETURNS MM WORDS OF BCD DATA
C                             STARTING AT FIRST NON-BLANK WORD
C                             A WORD IS DEEMED TO END ON A BLANK, A COMMA,
C                             OR THE TENTH CHARACTER
C
C     ENTRY     INREW     RESETS SO THAT NEXT CALL OF INDEC OR ININT
C                             WILL START ON NEW CARD
C
C                             NOTE ... NORMALLY DATA CARDS ARE READ FROM
C                                      5 AND PRINTED ON 6
C     ENTRY     INLIN     CHANGES INPUT UNIT TO MM
C     ENTRY     INLOU     CHANGES OUTPUT UNIT TO MM
C
000005        DIMENSION KARD(8),ARG(MM)                                    69.0709
000005        DATA(KOUNT=0),(K1=0),(IP= 5),(IO= 6),(MULT=0),(NCPW=10),(KAR=0)  69.0709
000005        EQUIVALENCE (AEG,IEG)                                        69.0709
C
000005        INTGER=0    $   GO TO 10
000006        ENTRY ININT
000015      9 INTGER=1
000016     10 DO 40 M=1,MM
000020        IF(KOUNT.GT.K1) GO TO 20
000023     12 READ(IP,1) KARD
000031      1 FORMAT(8A10)                                                 69.0709
000031        IF(CHKFIL(IP)) RETURN                                        69.0131
000035        IF(IO.EQ.0) CALL COUNOUT(1)                                  69.0131
000043        IF(IO.GT.0) WRITE(IO,3) KARD                                 69.0131
000054      3 FORMAT(5X,8A10)                                              69.0709
000054        KOUNT=80
000056        K1=0
C
000056     20 NDIG=NDOT=NMIN=NPLS=NEE=0
000063     22 K1=K1+1    >    IF(K1.GT.80) GO TO 12
000072        IF(INTGER.LT.0) GO TO 50
000073        CALL PCHAR(K1,KARD,NCPW,KAR,1)                               69.0709
000077        IF(KAR.EQ.1H-) NMIN=1
000102        IF(KAR.EQ.1H.) NDOT=1
000105        IF(KAR.GE.1H0 .AND. KAR.LE.1H9) NDIG=1                       68.0209
000117        IF(NMIN+NDOT+NDIG.LE.0) GO TO 22
000124        K2=K1
000125        MINF #ST=NMIN                                                68.0209
```

```
000371      50 CALL MCHAR(K1,KARD,NCPW,KAR,1)                                  69.0709
000375         IF(KAR.EQ.1H .OR. KAR.EQ.1R) GO TO 49                           69.0709
000406         DO 52 K=2,NCPW                                                  69.0709
000407         K2=K1+K-1
000411         IF(K2.GT.80) GO TO 54
000415         CALL MCHAR(K2,KARD,NCPW,KAR,1)                                  69.0709
000420      52 IF(KAR.EQ.1H .OR. KAR.EQ.1H ) GO TO 54                          69.0709
000433         K2=K1+NCPW                                                      69.0709
000435      54 K2=K2-1
000437         IEQ=1H                                                          69.0709
000440         CALL MCHAR(K1,KARD,1,IEQ,K2-K1+1)                               69.0709
000446         K1=K2
000450         GO TO 46
               C
               C
000452         ENTRY INLIN
000461      44 IM=MM  $  GO TO 90
000463         ENTRY INLOG
000472      48 IO=MM  $  GO TO 90
000474         ENTRY INREW
000503      90 KOUNT=0
000504      92 MULT=0
000505      99 RETURN
               C
000506         END  INDEC
               C
000126      24 K2=K2+1    $   IF(K2.GT.80) GO TO 34
000133         CALL MCHAR(K2,KARD,NCPW,KAR,1)                                  69.0709
000136         IF(KAR.LT.1H0.OR.KAR.GT.1H9) GO TO 26                           68.0209
000147         NDIG=NDIG+1
000150         GO TO 24
000151      26 IF(KAR.NE.1H.) GO TO 28
000153         NDOT=NDOT+1    $   IF(NDOT.GT.1) 34,24
000161      28 IF(KAR.NE.1HE) GO TO 30
000163         NEE=NEE+1    $   IF(NEE.GT.1) 34,24
000171      30 IF(KAR.NE.1H+) GO TO 32
000173         NPLS=NPLS+1    $   IF(NPLS+NMIN-MINF RST,GT.1) 34,24            68.0209
000203      32 IF(KAR.NE.1H-) GO TO 34
000205         NMIN=NMIN+1    $   IF(NMIN=MINF RST.GT.1) 34,24                 68.0209
               C
000214      34 IF(NDIG.LE.0) GO TO 20
000216         IF(MULT.GT.0 .OR. KAR.NE.1H ) GO TO 36
000225         IF(NDOT+NEE+NPLS+NMIN.EQ.0) MULT=-1
000231      36 K2=K2-1    $   KK=K2-K1+1
000235         FMT=4H(   F   )                                                 69.0709
000236         IF(K1.EQ.1) GO TO 37                                            69.0709
000237         L=(K1-1)/10  $  CALL MCHAR(NCPW,L+1H0,2,FMT,1)                  69.0709
000247         L=K1-1-10*L  $  CALL MCHAR(NCPW,L+1H0,3,FMT,1)                  69.0709
000260                         CALL MCHAR(NCPW,  1HX,4,FMT,1)                  69.0709
000264      37 L=KK/10     $  CALL MCHAR(NCPW,L+1H0,6,FMT,1)                   69.0709
000274         L=KK-10*L   $  CALL MCHAR(NCPW,L+1H0,7,FMT,1)                   69.0709
000304         DECODE(K2,FMT,KARD) AEW                                         69.0709
000316         IF(INTGER.GT.0 .OR. MULT.LT.0) IEQ=AEW
000327         K1=K2                                                           69.0709
000331         IF(MULT) 38,46,40                                               69.0709
000334      38 KMUL=K1=K1+1
000337         MULT=IEQ-1
000340         GO TO 20
000341      40 IF(KAR.NE.1H ) GO TO 46
000343         K1=KMUL
000345         MULT=MULT-1
000346      46 ANG(M)=AEW
000351      48 CONTINUE
000353         GO TO 99
               C
000353         ENTRY INALF
000362      47 INTGER=-1
000363         GO TO 10
000364      49 K1=K1+1
000366         IF(K1.GT.80) GO TO 12
```

```
                FUNCTION INDEQU (LENTH, MARAY, MATCH)
           *    SOURCE DATE 68.0209       CONVERT TO CDC 6600  -  BRAND NEW CODE
           C
           C    FOR ENTRY INDEQU, RETURNS INDEX I SUCH THAT MARAY(I)=MATCH
           C    FOR ENTRY INDMAX, RETURNS INDEX I SUCH THAT MARAY(I) IS LARGEST
           C    FOR ENTRY INDMIN, RETURNS INDEX I SUCH THAT MARAY(I) IS SMALLEST
           C
000005          DIMENSION MARAY(LENTH)
           C
000005       10 DO 12 L=1,LENTH
000007          IF (MARAY(L).EQ.MATCH) GO TO 14
000011       12 CONTINUE
000013          L=0
000014       14 INDEQU=L
000016          RETURN
           C
000016          ENTRY INDMIN
000025       20 LL=0  $  IF(LENTH-1) 26,21,23
000031       21 LL=1  $  GO TO 26
000033       23 MVAL=MARAY(1)
000036          LL=1
000037          DO 24 L=2,LENTH
000041          IF(MARAY(L).GE.MVAL) GO TO 24
000044       22 MVAL=MARAY(L)
000050          LL=L
000051       24 CONTINUE
000054       26 INDEQU=LL
000056          RETURN
           C
000056          ENTRY INDMAX
000065       30 LL=0  $  IF(LENTH-1) 36,31,33
000071       31 LL=1  $  GO TO 36
000073       33 MVAL=MARAY(1)
000076          LL=1
000077          DO 34 L=2,LENTH
000101          IF(MARAY(L).LE.MVAL) GO TO 34
000104       32 MVAL=MARAY(L)
000110          LL=L
000111       34 CONTINUE
000114       36 INDEQU=LL
000116          RETURN
000116          END  INDEQU
```

```
      SUBROUTINE INDO(TITLE,LOOPS,KASE,VALUES)                          1240
C     SOURCE DATE 68.0717    FIX LOOPING IF DOCON(4, ) = BLANK
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.0726   CALL OLDATA ON READING SPECIAL DATA-CARD
C     SOURCE DATE 67.0726   BRAND NEW CODE
C     SOURCE DATE 66.0601  CALL OUTSET WITH 300 IN FIRST ARG
C                                                                       1241
C         INPUTS, OUTPUTS, AND INCREMENTS UP TO 10 GENERALIZED DO LOOPS 1242
C                                                                       1243
C         TITLE  -  HOLLERITH TITLE FOR OUTPUT                          1244
C         LOOPS  -  NUMBER OF LOOPS TO BE INCREMENTED                   1245
C         KASE   -  NUMBER OF TRIAL IN PROGRESS                         1246
C         VALUES -  LIST OF VALUES OF INCREMENTED VARIABLES             1247
C                                                                       1248
C         FOR KASE=-1, INDO READS AND PRINTS FOR EACH LOOP A CARD       1249
C         SPECIFYING LOOP INCREMENTATION -                              1250
C                                                                       1251
C              COLUMNS   FORMAT    CONTENT                            1252
C              -------   ------    -------                            1253
C              1-30      3A8,A6    HOLLERITH IDENTIFICATION OF LOOP   1254
C                                  VARIABLE                           1255
C              31-40     F10.0     MINIMUM VALUE ASSUMED BY VARIABLE  1256
C              41-50     F10.0     MAXIMUM VALUE ASSUMED BY VARIABLE  1257
C              51-60     F10.0     VALUE SELECTOR FOR VARIABLE        1258
C              61-70     F10.0     SELECTION METHOD CODE              1259
C                                                                       1260
C         SELECTION METHODS ARE DEFINED AS FOLLOWS -
C
C              METHOD    VALUE     SELECTION
C              CODE      SELECTOR  METHOD
C              -----     --------  ----------------------------
C
C              0 OR 1  INCREMENT   LINEAR INCREMENTATION
C
C                           EXAMPLES ...
C                            FOR LOOP FROM 1 TO 3 IN STEPS OF .5
C                                PUNCH   1.  3.  .5  1.
C                            FOR LOOP FROM 3 TO 1 IN STEPS OF .5
C                                PUNCH   1.  3.  -.5  1.
C
C              2       STEPS/      LOGARITHMIC INCREMENTATION
C                      DECADE
C                           EXAMPLES ...
C                            FOR LOOP FROM 3 TO 300 AT 4 STEPS/DECADE
C                                PUNCH   3.  300.  4.  2.
C                            FOR LOOP FROM 300 TO 3 AT 4 STEPS/DECADE
C                                PUNCH   3.  300.  -4.  2.
C
C              3     (INCREMENT)  RANDOM SELECTION
C
C                           IF VALUE SELECTOR IS ZERO, VALUES ARE
C                           TAKEN AT RANDOM BETWEEN MIN AND MAX VALUES
C                           EXAMPLE ...
C                            FOR RANDOM VARIABLE BETWEEN 5 AND 6
C                                PUNCH   5.  6.  0.  3.
C                           IF VALUE SELECTOR IS NON-ZERO, VALUES ARE
C                           PICKED RANDOMLY FROM THE SET OF DISCRETE
C                           VALUES WHICH WOULD OCCUR UNDER METHOD 1
C                           EXAMPLE ...
C                            FOR RANDOM INTEGERS BETWEEN 1 AND 10
C                                PUNCH   1.  10.  1.  3.
C
C         A VARIABLE SELECTED BY METHOD 3 IS NOT, STRICTLY SPEAKING,
C         IN A LOOP - THE SEQUENCE DOES NOT TERMINATE ITSELF, BUT MUST
C         BE STOPPED BY THE SATISFACTION OF OTHER (TRUE) LOOPS, OR BY
C         DISCONTINUING THE CALLS OF INDO.
C
```

```
C         VARIABLES SELECTED BY METHODS 0, 1, OR 2 ARE ORDINARY LOOP
C         VARIABLES - THEY PROGRESS FROM ONE EXTREME VALUE TO THE OTHER.
C         IF THE METHOD SELECTION CODE IS POSITIVE, THE LOOP WILL BE
C         NESTED - THE OUTER-LOOP (EARLIER-READ) VARIABLES WILL NOT
C         CHANGE UNTIL THIS (INNER) LOOP IS SATISFIED. FOR NEGATIVE
C         METHOD CODES, THIS RESTRAINT IS REMOVED, ALLOWING SIMULTANEOUS
C         STEPPING. EXAMPLE ...
C              DATA-CARDS PUNCHED  (A)  1.   2.   .5   1.
C                                  (B)  10.  100.  2. -2.
C              RESULTING SEQUENCE  (A,B) = (1.,10.),(1.5,31.6),(2.,100.)
C
C         WHEN LOOPS ARE NESTED, THE LAST-READ LOOP IS INNERMOST
C                                                                        1278
C         FOR KASE GREATER THAN OR EQUAL TO ZERO, INDO INCREMENTS KASE   1279
C         BY ONE, SELECTS NEW VALUES AS REQUIRED.  IF ALL LOOPS ARE THUS 1280
C         SET AT FINAL VALUES, INDO RESETS KASE TO 0.                    1281
C                                                                        1282
C
C         TYPICAL USE IS AS FOLLOWS -
C
C         1 KASE=-1
C           CALL INDO(TITLE,LOOPS,KASE,VALUES)
C                (TO READ DATA-CARDS)
C         2 CALL INDO(TITLE,LOOPS,KASE,VALUES)
C                (TO SET UP VALUES)
C           IF(KASE.EQ.0) GO TO 1
C           IF(KASE.GE.MAX) STOP
C           (COMPUTE, PRINT, ETC)
C           GO TO 2
C
C
C         IF TITLE IS AN ASTERISK FOLLOWED BY BLANKS, ALL OUTPUT IS      1283
C         DELETED.                                                       1284
C                                                                        1285
      CCOMPKG,BASCON                                                  67.0726
000006        COMMON /BASCON/
     1   PROGRM,    KPAGE,     LINE,      TOFDAY,    RUN,       RUNID(6),
     2   MSG,       FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
     3   KLASS,     KGROUP,    KUNITS,    KSTINT,    KOORD,     IFEOF
     4  ,NDFILE
C
C                                                                        1287
000006        DIMENSION VALUES(2),H(82),NAMES(5,10),DOCON(4,10)          1288
000006        DIMENSION VFIRST(10),MODE(10),AA(10),BB(10),NTM(10),NTIM(10)
000006        DATA (ASTRSK=1H*)
000006        DIMENSION DOC(40)                                       67.0726
C
000006        IF(KASE) 100,200,300
000007    100 CALL OUTSET(344,12H  MINIMUM    ,12H   VALUE     ,12H            ,H) 66.0601
000013        CALL OUTSET( 54,12H  MAXIMUM    ,12H   VALUE     ,12H            ,H)    1293
000017        CALL OUTSET( 64,12H     VALUE   ,12H  SELECTOR   ,12H            ,H)    1294
000023        CALL OUTSET( 74,12H SELECTION   ,12H   METHOD    ,12H            ,H)    1295
000027        CALL INCOL(TITLE,H,NAMES,DOC,LOOPS)                     67.0726
000035        IF(NDFILE) RETURN                                       67.0726
000037        IF(NAMES(1,1) .NE.6HOLD DATA) GO TO 102                 67.0726
000044        CALL OLDATA(4HINDO)                                     67.0726
000045        GO TO 104                                               67.0726
000050    102 CALL XMIT(40,DOC,DOCON)                                 67.0726
000053    104 CALL OUTSET(354,12H             ,12H             ,H)
000057        DO 106 L=1,LOOPS                                        67.0717
000063        IF(DOCON(4,L).EQ.-0.) DOCON(4,L)=0.                     67.0717
000071    106 CONTINUE                                                67.0717
000074        KASE=0                                                     1297
000074        RETURN                                                     1298
C
000075    200 ASSIGN 236 TO KUMBAK
000076        DO 236 LP=1,LOOPS
000100        DV=DOCON(3,LP)
000103        AV=ABS(DV)
000105        DD=DOCON(2,LP)-DOCON(1,LP)
000111        VV=DOCON(1,LP)
000114        IF(DV.LT.0.) VV=DOCON(2,LP)
000117        VFIRST(LP)=VV
000122        MODE(LP)=ABS(DOCON(4,LP))
000126        IF(MODE(LP).EQ.0) MODE(LP)=1
000132        M=MODE(LP)                                              68.0209
000135        GO TO (210,220,230),M                                   68.0209
000143    210 AA(LP)=DV
000146        BB(LP)=1.
000150        NTM(LP)=FDIV(DD/AV)+.999                                68.0209
000157        GO TO 222
000162    220 AA(LP)=0.
```

```
000164          BB(LP)=10.**FDIV( 1./DV)                                        68.0209
000176          NTM(LP)=ALOG10(FDIV(DOCON(2,LP)/DOCON(1,LP)))*AV+.999           68.0209
000214      222 VALUES(LP)=VV
000221          GO TO 236
000222      230 IF(DV.NE.0.) GO TO 232
000223          AA(LP)=DOCON(1,LP)
000227          BB(LP)=DD
000231          GO TO 234
000232      232 AA(LP)=VV
000235          BB(LP)=FDIV(DD/AV)                                              68.0209
000243      234 NTM(LP)=0
000245          GO TO 500
000250      236 NTIM(LP)=ISIGN(NTM(LP),DOCON(4,LP))
000263          GO TO 900
            C
000263      300 ASSIGN 304 TO KUMBAK1
000264          ASSIGN 304 TO KUMBAK
000265          DO 306 LP=1,LOOPS
000267          LL=LP+1
000271          IF(LL.GT.LOOPS) GO TO 303
000274          DO 302 L=LL,LOOPS
000275          IF(NTM(L).GT.0 .AND. DOCON(4,L ).GT.0.) GO TO 306
000310      302 CONTINUE
000312      303 DV=DOCON(3,LP)
000316          IF(MODE(LP).EQ.3) GO TO 500
000320          IF(NTM(LP).NE.0) GO TO 400
000322          VALUES(LP)=VFIRST(LP)
000324          NTM(LP)=NTIM(LP)
000327          GO TO 306
000327      304 NTM(LP)=ISIGN(IABS(NTM(LP))-1,NTM(LP))
000335      306 CONTINUE
000340          GO TO 900
            C
000340      400 VALUES(LP)=VALUES(LP)+BB(LP)*AA(LP)
000350          GO TO KUMBAK1
            C
000353      500 IF(DV.NE.0.) GO TO 502
000354          VALUES(LP)=RANF(00)*BB(LP)+AA(LP)
000366          GO TO KUMBAK
000372      502 VALUES(LP)=DV*AINT(RANF(00)*AINT(BB(LP)+1.))+ AA(LP)            68.0209
000413          GO TO KUMBAK
            C
000417      900 DO 902 LP=1,LOOPS
000421          IF(NTM(LP).GT.0) GO TO 904
000424      902 CONTINUE
000426          KASE=-1
000427      904 KASE=KASE+1
000433       90 IF (TITLE.NE.ASTRSK) CALL OUTCOL(TITLE,M,NAMES,VALUES,LOOPS)     1348
000443          RETURN
            C
000444          END  INDO
```

```
      SUBROUTINE INMV (TITLE,MODE,VALUE)
C     SOURCE DATE 69.0709    REMOVE CALL OF AMIXER
C     SOURCE DATE 68.1205   ADD COMMON /BASCON/
C     SOURCE DATE 68.0724    BRAND NEW CODE
C
C     INMV INPUTS A MODE AND A VALUE FROM A DATA CARD
C     TITLE IS A BCD CHARACTER STRING ENDING ON A DASH
C
C     CARD FORMAT ...
C        COL 1-41  ANY COMMENTS
C        COL 41-50 MODE, PUNCHED IN F-FORMAT
C        COL 51-60 VALUE
C        COL 61-63 UNITS OF VALUE, PUNCHED IN A-FORMAT
C     VALUE WILL BE SCALED TO INTERNAL UNITS AS SHOWN BELOW ...
C          UNIT-NAME PUNCHED   INTERNAL (MET.)    (ENGL.)
C                    ----------        ----------     --------
C                    DEG               RADIANS        RADIANS
C                    FT                METERS         FEET
C                    KM                METERS         FEET
C                    NMI               METERS         FEET
C                    KFT               METERS         FEET
C                    M                 METERS         FEET
C
CCOMPKG,BASCON
000006       COMMON /BASCON/
            1   PROGNM,    KPAGE,     LINE,      TOFDAY,    RUN,       RUNID(6),
            2   MSG,       FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
            3   KLASS,     KGROUP,    KUNITS,    KSIINT,    KOURD,     IFEOF
            4  ,NDFILE
CCOMPKG,BASCON                                                          09/15/69
C
CCOMPKG,CONCON
000006       COMMON /CONCON/
            1   PI,        SRD,       SLV,       SMF,       SKM,       RBODY,
            2   GACC,      GCON,      WBODY,     RHOZRO,    TWOPI,     HAFPI
CCOMPKG,CONCON                                                          09/15/69
C                                                                          69.0709
CCOMPKG,CONVRT                                                             69.0709
000006       COMMON /CONVRT/ NNV, NAMVAL(3,11)                             69.0709
000006       DATA (NNV=11)                                                 68.0527
000006       DATA (NAMVAL =                                                68.0527
            1   3HDEG,  2(.01745329252),                                  68.0527
            2   2HFT,   0.3048006,  1.0,                                  68.0527
            3   3HPSF,  4.88024,  1.0,                                    68.0527
            4   2HKM,   1000.,  3280.833,                                 68.0527
            5   3HNMI,  1853.25,  6080.3,                                 68.0527
            6   2HLB,   0.453592,  1.0,                                   68.0527
            7   1HG,    9.80665,  32.17398,                               68.0527
            8   3HKFT,  304.8006,  1000.,                                 68.0527
            9   1HM,    1.0,   3.280833 ,                                 69.0709
            0   3HMHA,  2(.001) ,                                         69.0709
            1   3HSEC,  2(1.)  )                                          69.0709
CCOMPKG,CONVRT                                                             69.0709
C                                                                       09/15/69
000006       DIMENSION             TEXT(4)                                 69.0709
C
000006       DATA (AND=1H )
```

```
000006          EQUIVALENCE (AEQ,IEQ)                                   69.0709
      C
000006          CALL TITLEH(-1,TITLE)
000010          READ(5,1) TEXT,ODE,VALU,NAM,NAM2                        69.0709
000026        1 FORMAT(4A10,2F10,A3,A7)                                 69.0709
000026          IF(CHEKFIL(5)) RETURN
000032          DO 10 N=1,NNV                                           69.0709
000036          IF(NAM.EQ.NAMVAL(1,N)) GO TO 12
000041       10 CONTINUE
000043          VALUE=VALU
000044          GO TO 14
000045       12 IEQ=NAMVAL(KUNITS+2,N)                                  69.0709
000051          VALUE=VALU*AEQ                                          69.0709
000052       14 MODE=ODE
000054          CALL COUNOUT(1)
000055          IF(VALUE.EQ.VALU) GO TO 18
000081          WRITE(6,2) TEXT,MODE,VALU,NAM,NAM2,AND,VALUE            69.0709
000102        2 FORMAT(15X,4A10,1H,,3XF10.3,3XA3,A7,A3,*(*G13.6,* AFTER SCALING)*)  69.0709
000102          RETURN
      C
000103       18 WRITE(6,2) TEXT,MODE,VALU,NAM,NAM2                      69.0709
000123          RETURN
000124          END  INMV
```

```
      SUBROUTINE JACOBI(A,B,N,ACCU,IV,NR)
C     SOURCE DATE 66.1205  MAKE IT WORK FOR DIAGONAL MATRICES, TOO
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 64.0701   BRAND NEW CODE
CJACOBI     SUBROUTINE FOR BMD03M      7-1-64                          M0303400
C
C     A IS A REAL SYMMETRIC MATRIX
C     B IS A DUMMY IF IV=0.  IF IV=1, IS LOCATION WHERE EIGENVECTOR
C     MATRIX IS CREATED, AND IF IV=2 IS A MATRIX WHICH WILL BE
C     PRE-MULTIPLIED BY THE EIGENVECTOR MATRIX.
C     N IS SIZE OF A AND B
C     ACCU IS A PRECISION VALUE.  IF LT 1.E-10, WILL FORCE TO 1.E-10.
C     IV IS A SWITCH WHICH MAY TAKE ON VALUES 0, 1, OR 2.  SEE B
C     PARAMETER DESCRIPTION ABOVE.
C     NR IS AN ITERATION COUNTER.
C
C
000011      DIMENSION A(N,N), B(N,N), LK(160), Q(160)
C
000011      ACC=AMAX1F(ACCU,1.E-10)
000014      IF(IV.NE.1) GO TO 200                                       68.0209
000016      CALL XMIT(-N,1.,Q)                                          68.0209
000021      CALL MATDIAG (Q,B,N)                                        68.0209
C
000026  200 NR=0                                                        M0303480
000032      B=0.                                                        M0303490
000033      W=0.                                                        M0303500
000034      MW=.5*A*A                                                   M0303510
000036      II=0                                                        68.1205
000037      DO 1 I=2,N                                                  M0303520
000040      MW=MW+.5*A(I,I)*A(I,I)                                      M0303530
000046      Q(I)=0.                                                     M0303540
000050      I1=I-1                                                      M0303550
000051      DO 2 J=1,I1                                                 M0303560
000053      Y=ABSF(A(I,J))                                              M0303570
000060      MW=MW+Y*Y                                                   M0303580
000062      IF(Y-Q(I))2,2,3                                             M0303590
000065    3 Q(I)=Y                                                      M0303600
000070      LK(I)=J                                                     M0303610
000072    2 CONTINUE                                                    M0303620
000075      IF(Q(I)-W)1,1,4                                             M0303630
000100    4 W=Q(I)                                                      M0303640
000103      III=I                                                       M0303650
000104    1 CONTINUE                                                    M0303660
000107      IF(III.EQ.0) GO TO 31                                       68.1205
C
000110      MW=ACC*SQRTF(2.*MW)/FLOATF(N)                               M0303670
C
000123   30 II=LK(III)                                                  M0303680
000126      JJ=III                                                      M0303690
000127      V=A(III,II)                                                 M0303700
000133      Y=A(JJ,II)                                                  M0303710
000137      Y=A(JJ,JJ)                                                  M0303720
000142      W=X-Y                                                       M0303730
000144      T=.5*FDIV((W*SQRTF(W*W+4.*V*V))/V)
000157      W=SQRTF(1.+T*T)                                             M0303750
000164      S=FDIV(T/W)
000170      C=FDIV(1./W)
000174      CC=C*C                                                      M0303780
000175      SS=S*S                                                      M0303790
000176      SC=S*C*2.                                                   M0303800
000200      Q1=0.                                                       M0303810
000201      Q2=0.                                                       M0303820
000202      W=0.                                                        M0303830
000203      NR=NR+1                                                     M0303840
C
000210      DO 27 I=1,N                                                 M0303850
000211      IF(I-II)10,11,12                                            M0303860
000213   10 II=A(II,I)                                                  M0303870
000220      V=A(JJ,I)                                                   M0303880
000224      F=U*S+V*C                                                   M0303890
000227      A(II,I)=E                                                   M0303900
000233      IF(ABSF(E)-Q1)15,15,14                                      M0303910
000236   14 Q1=ABSF(E)                                                  M0303920
000240      I1=I                                                        M0303930
000242   15 F=V*S-U*C                                                   M0303940
```

```
000246          A(JJ,I)=F                                                       M0303950
000253          IF(ABSF(F)-Q2)9,9,16                                            M0303960
000256    16    Q2=ABSF(F)                                                      M0303970
000260          I2=I                                                            M0303980
000262          GO TO 9                                                         M0303990
                C
000262    11    A(II,I)=SS*X+SC*Y+CC*Z                                          M0304000
000274          Q(I)=Q1                                                         M0304010
000276          LK(I)=I1                                                        M0304020
000300          GO TO 9                                                         M0304030
                C
000301    12    IF(I-JJ)17,18,19                                                M0304040
000304    17    U=A(I,II)                                                       M0304050
000311          V=A(JJ,I)                                                       M0304060
000315          E=SS*U-CC*V                                                     M0304070
000320          A(I,II)=E                                                       M0304080
000325          IF(ABSF(E)-Q(I))15,15,21                                        M0304090
000330    21    LK(I)=II                                                        M0304100
000333          Q(I)=ABSF(E)                                                    M0304110
000336          GO TO 15                                                        M0304120
000336    18    A(JJ,I)=CC*X-SC*Y+SS*Z                                          M0304130
000350          A(I,II)=0.                                                      M0304140
000354          Q(I)=Q2                                                         M0304150
000356          LK(I)=I2                                                        M0304160
000360          GO TO 9                                                         M0304170
000361    19    U=A(I,II)                                                       M0304180
000366          V=A(I,JJ)                                                       M0304190
000372          E=U*S+V*C                                                       M0304200
000375          F=V*S-U*C                                                       M0304210
000377          A(I,II)=E                                                       M0304220
000404          A(I,JJ)=F                                                       M0304230
000410          Q=MAX1F(ABSF(E),ABSF(F))                                        M0304240
000417          IF(Q-Q(I))9,9,13                                                M0304250
000423    13    Q(I)=Q                                                          M0304260
000426          IF(ABSF(E)-ABSF(F))23,24,24                                     M0304270
000432    24    LK(I)=II                                                        M0304280
000435          GO TO 9                                                         M0304290
000435    23    LK(I)=JJ                                                        M0304300
000440    9     IF(Q(I)-W)40,25,25                                              M0304310
000444    25    W=Q(I)                                                          M0304320
000447          III=I                                                           M0304330
000450    40    IF(IV)27,27,33                                                  M0304340
000452    33    U=B(I,II)                                                       M0304350
000457          V=B(I,JJ)                                                       M0304360
000463          B(I,II)=U*S+V*C                                                 M0304370
000471          B(I,JJ)=V*S-U*C                                                 M0304380
000477    27    CONTINUE                                                        M0304390
000502          IF(W-H)31,31,30                                                 M0304400
000504    31    IF(IV)55,55,56                                                  M0304410
000506    56    DO 50 I=1,N                                                     M0304420
000510          U=-1.E20                                                        M0304430
000512          DO 51 J=I,N                                                     M0304440
000513          IF(A(J,J)-U)51,51,52                                            M0304450
000520    52    U=A(J,J)                                                        M0304460
000524          K=J                                                             M0304470
000525    51    CONTINUE                                                        M0304480
                C
000530          IF(K-I)53,50,53                                                 M0304490
000532    53    A(K,K)=A(I,I)                                                   M0304500
000541          A(I,I)=U                                                        M0304510
000545          DO 54 J=1,N                                                     M0304520
000546          U=B(J,K)                                                        M0304530
000552          B(J,K)=B(J,I)                                                   M0304540
000560    54    B(J,I)=U                                                        M0304550
000565    50    CONTINUE                                                        M0304560
                C
                C     THIS CHANGE FORCES FIRST NON-ZERO ELEMENT OF EACH EIGENVECTOR POSITIVE.
000570       55 IF (IV.LE.0) RETURN
                C
000573       60 DO 70 J = 1,N
000575          DO 62 I = 1,N
000576          IF (B(I,J)) 64,62,70
000603       62 CONTINUE
000606       64 DO 66 I = 1,N
000610       66 B(I,J) = -B(I,J)
000617       70 CONTINUE
                C
000622          RETURN
000622          END  JACOBI
```

```
VER 1.1       KALLER                                                    08/30/69      PAGE NO.      1

                                        IDENT     KALLER
                          000012    PROGRAM LENGTH

                                    BLOCKS

                    000000    000012    PROGRAM*   LOCAL

                                    ENTRY POINTS

                                       000001 KALLER

                                    *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                                          ENTRY     KALLER
000000  1301141405220000000l              VFD       36/6LKALLER,24/1
000001                                    *
                                          *         WHEN CALLED WITH ARGUMENT N, RETURNS NAME OF N-TH ROUTINE
                                          *         IN THE STRING OF CALLING ROUTINES, SUCH THAT ZERO
                                          *         CORRESPONDS TO THE ROUTINE THAT CALLED KALLER.
                                          *         ALSO, RETURNS REL ADDR OF CALL IN ARGUMENT N
                                          *
000001                                    KALLER    BSSZ      1
000002  6170000001                        LOOP      SB7       1
                    67307                           SB3       -B7
000003  6140000034                                  SB6       30
                    6150000077                      SB5       770
000004  56110                                       SA1       B1
                    63210                           SB2       X1
                              7130000000 *          SX3       KALLER-1
000005  53237                                       SA2       X3+B7       FETCH JUMP-BACK INSTRUCTION
                    22262                           LX2       B6,X2       CIRC SHIFT ... RET ADDR NOW IN LOWER END
                          53323                     SA3       X2+B3       FETCH RETURN-JUMP INSTRUCTION
                                53430               SA4       X3          FETCH NAME/NARGS
000006  63440                                       SB4       X4          NUMBER OF ARGUMENTS
                    0754000010 *                    LT        B5,B4,DONE
                                67227               SB2       B2-B7       DECREMENT COUNTER
000007  0620000005 *                                PL        B2,LOOP
000010  10644                               DONE      BX6       X4          PUT NAME IN X6
                    64440                           SB4       A4
                          75734                     SA7       A3-B4       FIND REL ADDR
                                56710               SA7       B1          STORE IN ARGUMENT N
000011  0200000001 *                                JP        KALLER
000012                                              END

                          034513    UNUSED STORAGE          32 STATEMENTS       3  SYMBOLS
```

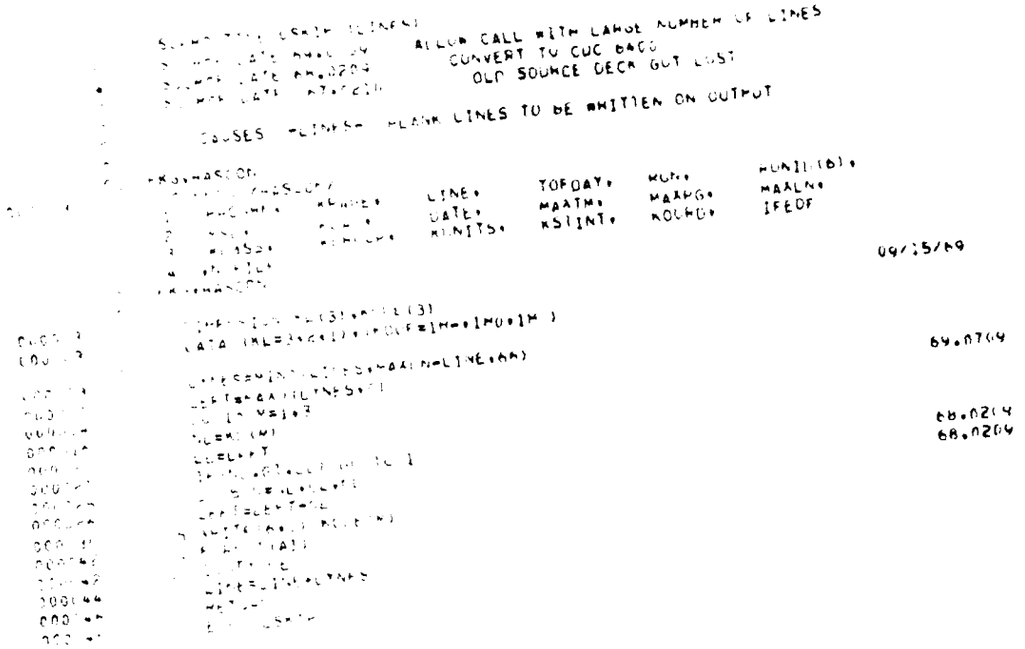
```
      FUNCTION KONVERG (ERROR,VARBL,WORK)
C     SOURCE DATE 69.0723   SPECIAL HANDLING FOR HUGE VALUES OF ERROR
C     SOURCE DATE 69.0709    ASSURE PRCSN POSITIVE, ADD ENTRY SETVERG
C     SOURCE DATE 68.0814   INITIALIZE TO ZEROS
C     SOURCE DATE 67.1205   ADAPTED FROM DIVERGE OF 67.0714
C
C     VARBL IS RESET, TO DRIVE ERROR TO ZERO,,, KONVERG RETURNS A FLAG
C     TO INDICATE PROGRESS OF CONVERGENCE.
C
C     SELECTION OF STEP IN VARBL IS AS FOLLOWS ...
C        IF ERROR AND ELAST BOTH HUGE, REPEAT INITIAL STEP
C        IF EITHER ERROR OR ELAST HUGE (NOT BOTH), HALVE THE INTERVAL
C        IF BOTH ERROR AND ELAST ARE REASONABLE, USE SLOPE THRU LAST 2 POINTS
C
C    KONVERG  MUST BE INITIALIZED AS FOLLOWS .....
C     CALL KONVSET  (DV,PRCSN,WORK)
C  OR CALL SETVERG  (DV,PRCSN,WORK)      WHERE ....                          69.0709
C                      DV IS INITIAL INCREMENT TO BE USED ON VARBL
C                      PRCSN IS ALLOWABLE TOLERANCE ON VARBL                69.0709
C                      WORK IS 8-WORD WORK-SPACE USED BY KONVERG
C
C    KONVERG  RETURNS FUNCTION VALUE =  0 IF CONVERGENCE PROCEEDING NORMALLY
C                                      +1 IF CONVERGENCE COMPLETED
C                                      -1 IF FAILURE TO CONVERGE
C
C     TYPICAL USE IS AS FOLLOWS  .....
C        DIMENSION WORK(8)
C        CALL KONVSET  (10.0,.OOOOI,WORK)
C        V=100.
C  10    TEST=FCN(V) - GOAL
C        IF(KONVERG (TEST,V,WORK)) 20,10,30
C  20    (ERROR MESSAGE)
C  30    (PROCEED)
C
C
000006        EQUIVALENCE (AEW,IEW)
000006        DIMENSION WORK(8),WRK(8)
00000A        EQUIVALENCE (WRK(1), KOUNT ),(WRK(2), PRCSN )
000006        EQUIVALENCE (WRK(3), VNOW  ),(WRK(4), VLAST )
00000A        EQUIVALENCE (WRK(5), ENOW  ),(WRK(6), ELAST )
00000A        EQUIVALENCE (WRK(7), DV    ),(WRK(8), DVLAST )
000006        DATA (LIMIT=99),(KOUNT=-1)                                      69.0723
C
00000A        IF(KOUNT.LT.0 .OR. WORK(1).LT.0.)  CALL CEASE                   69.0709
     *  (46H- KONVERG HAS FAILED, OR WAS NOT INITIALIZED -)                   69.0709
000020        CALL XMIT(8,WORK,WRK)
000023        KOUNT=KOUNT+1
000025        KFUNC=1
00002A        IF(KOUNT.GT.LIMIT-2) KFUNC=2                                    69.0723
000032        IF(KOUNT.GT.LIMIT) KFUNC=3
000035        IF(KOUNT.GT.2 .AND. ABS(FDIV(ERROR,ENOW)).GT.1.
     *                      .AND. ERROR*ENOW.GT.0.
     *                      .AND. ABS(FDIV(ENOW,ELAST)).GT.1.) KFUNC=4
000102        GO TO (20,10,10,10),KFUNC
C
000112    10 CALL COUNT OUT(8)
000114       WRITE(6,11) KOUNT,VLAST,ELAST,VNOW,ENOW,VARBL,ERROR
```

PRECEDING PAGE BLANK-NOT FILMED

```
000137      11 FORMAT(/5X,*TROUBLE CONVERGING ....  VARIABLE      ERROR  /
                A       * ITERATION *I3/
                X     3(E3*.3,E12.3/) )
000137        IF(KFLAG.LT.3) GO TO 12
000143        CALL FCLR FCTL(1)
000145        CALL WR ERROR(1,2HMCHAIN OF CALLING PROGRAMS -.)
000147      12 GO TO (42,20,40,50),KFUNC
            C
000151      20 VLAST=VNOW
000154        ELAST=ENOW
000156        VARBL=VARBL
000165        ENOW=ERROR
00016A        IF(ERROR.EQ.0.0) GO TO 99                                          69.0723
00016A        IF(NOITR.LE.1) GO TO 42                                            69.0723
000177        IF(ABS(ERROR).GT.1.E15 .AND. ABS(ELAST).GT.1.E15) GO TO 41         69.0723
000203        RATIO=ABS(FLIV(ERROR/ELAST))                                       69.0723
000211        IF(RATIO.GT.1.E-10 .AND. RATIO.LT.1.E+10) GO TO 40                 69.0723
000221        IF(ERROR*ELAST.GT.0.) GO TO 42                                     69.0723
000225        IF(ABS(DV).GT.PRCSN) GO TO 36                                      69.0723
000241        VARBL=VNOW + IF(RATIO.GT.1.E+10) VARBL=VLAST                       69.0723
000245        GO TO 99                                                           69.0723
000247      36 DV=DVLAST/2.                                                      69.0723
000251        GO TO 41                                                           69.0723
000253      40 DV=VLAST-VARBL*ENOW/(ERROR-ELAST)                                 69.0709
000266        IF(ABS(V/DVLAST) .GT. 10.  ) DV=SIGN(10.*DVLAST,DV)
000266      41 DVLAST=DV                                                         69.0723
000267        IF(ABS(DV).LT.PRCSN) KFUNC=3                                       69.0709
000283      42 VARBL=VARBL+DV
            C
000286      50 IF(KFLAG.EQ.2) 73                                                 69.0709
000311        IF(NOITR.GT.1) KOUNT=1
000316        KFLAG=KFUN
000326      55 CALL AMIII(3,2HM.,6HCHK)                                          69.0709
000277        RETURN
000301      99 KFUNC=3                                                           69.0723
000302        GO TO 91                                                           69.0723
            C
            C
            C
000305        ENTRY RSTVSET
000315        ENTRY SETVERR                                                      69.0709
000325        CALL AMIII(3,6HERR=,ERROR)                                         68.0816
000340        DVLAST=DV=ERROR
000346        PRCSN=ABS(VARBL)                                                   69.0709
000355        GO TO 95                                                           69.0709
000356        END  RSTVSET
```

```
VER 1.1      KSHFT                                                    08/30/68      PAGE NO.      1

                                        IDENT     KSHFT
                            000012      PROGRAM LENGTH

                                        BLOCKS

                    000000  000012      PROGRAM*  LOCAL

                                        ENTRY POINTS

                                           000001 KSHFT        000001 SHFT

                                        *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                                        *
                                        *         ARGUMENTS ... WORD,NBITS (+=RIGHT,-=LEFT)
                                        *         SHIFTS #ORD NBITS TO RIGHT OR LEFT, END OFF (ZERO FILL)
                                        *         THIS FUNCTION MAY BE CALLED BY EITHER NAME  #KSHFT# OR #SHFT#.
                                        *         RETURNS ##ORD# SHIFTED TO RIGHT OR LEFT WITH ZERO FILL.
                                        *

                                                  ENTRY     KSHFT
                                                  ENTRY     SHFT
                                                  VFD       30/5LKSHFT,30/2
                                        SHFT      BSS       0
                                        KSHFT     BSSZ      1
                                                  SA1       A1          #ORD IN X1
                                                  SA2       A2          NBITS IN X2
                                                  BX6       X1
                                                  ZR        X2,KSHFT    IF NBITS=0, RETURN ORIGINAL #ORD
                                                  SB3       X2          NBITS IN B3
                                                  SB4       B3
                                                  SB5       1
                                                  GE        B4,B0,RA
                                                  SB4       B0-B3
                                                  SA5       ONEBIT
                                                  SB6       B4-B5       MAKE MASK
                                                  AX5       BA,X5           IN
                                                  LX5       B4,X5       RIGHT END
                                                  UP        RA
                                        FA        SA5       ONEBIT
                                                  SB4       B4-B5       MAKE MASK IN
                                                  AX5       BA,X5       LEFT END
                                        RA        AX6       B3,X6
                                                  BX6       -X5*X6
                                                  UP        KSHFT
                                        ONEBIT    DATA      400000000000000000008
                                                  END

                                        NO ERRORS        35 STATEMENTS        5  SYMBOLS
```

```
                      SUBROUTINE LOCLAX (PVEC,QVEC,KP,KQ,UVEC)
                C     SOURCE DATE 67.0726   CALL CROS1 INSTEAD OF CROSS AND UNITV
                C     SOURCE DATE 66.0101   BRAND NEW CODE
                C
                C         RETURNS ORTHOGONAL UNIT-VECTORS IN UVEC, AS FOLLOWS
                C              UVEC(1-3,KP)   THE KP-TH VECTOR IS IN THE PVEC DIRECTION
                C              UVEC(1-3,KQ)   THE KQ-TH VECTOR IS NORMAL TO PVEC, AND
                C                             AS CLOSE AS POSSIBLE TO QVEC
                C              OTHER UVEC     COMPLETES THE RIGHT-HANDED SET
                C
                C         *** NOTE THAT UVEC IS DIMENSIONED (3,3)
                C
000006                DIMENSION PVEC(3),QVEC(3),UVEC(3,3),U(3,3)
                C
000006                CALL UNITV(PVEC,U(1,KP))
000013                CALL UNITV(QVEC,U(1,KQ))
000023                KK=6-KP-KQ
000030                KM=MOD(KQ-KP+3,3)
000035                GO TO (1,2),KM
000042              1 CALL CROS1(U(1,KP),U(1,KQ),U(1,KK))                              67.0726
000056                CALL CROSS(U(1,KK),U(1,KP),U(1,KQ))
000075                GO TO 3
000101              2 CALL CROS1(U(1,KQ),U(1,KP),U(1,KK))                              67.0726
000115                CALL CROSS(U(1,KP),U(1,KK),U(1,KQ))
000134              3 CALL XMIT(9,U,UVEC)
000142                RETURN
000143                END  LOCLAX
```



```
      SUBROUTINE MATADD (A,B,C,NRA,NCA)
C     SOURCE DATE 87.0726   BRAND NEW CODE
C
C      PERKINS, C. W.   MATRIX ADDITION AND SUBTRACTION
C
C      ENTRY MATADD  C= A + B
C      ENTRY MATSUB  C= A - B
C      NRA = NUMBER OF ROWS
C      NCA = NUMBER OF COLUMNS
C
 00006      DIMENSION A(NRA,NCA), B (NRA,NCA), C(NRA,NCA)
C
000006        DO 1  I=1,NRA
000010        DO 1  J=1,NCA
000011        C(I,J) = A(I,J) + B(I,J)
000026      1 CONTINUE
000033        RETURN
C
000033        ENTRY MATSUB
000044      3 DO 2  I=1,NRA
000046        DO 2  J=1,NCA
000047        C(I,J)=A(I,J)-B(I,J)
000064      2 CONTINUE
000071        RETURN
000071        END  MATADD
```

```
      C     SUBROUTINE MATDIAG  (V,A,N)
      C     SOURCE DATE 67.0726   #BRAND NEW CODE
      C
      C         STORES VECTOR V ON THE DIAGONAL OF SQUARE MATRIX A(N,N)
      C
000005        DIMENSION V(N),A(N,N)
000005
000012        CALL AMIT(-N**2,0.,A)
000016        DO 2 K=1,N
000027    2   A(K,K)=V(K)
000027        RETURN
000027        END  MATDIAG
```

PRECEDING PAGE BLANK-NOT FILMED

```
                      SUBROUTINE MATFLIP (A, NRA)
                C     SOURCE DATE 67.0726   BRAND NEW CODE
                C
                C     REPLACES SQUARE (NRA BY NRA) MATRIX -A- BY ITS TRANSPOSE
                C
                C     PERKINS, C. W.      27 APRIL 1967
                C
      000004          DIMENSION A(NRA, NRA)
                C
      000004          DO 1 I = 2, NRA
      000005          K = I-1
      000007          DO 1 J=1,K
      000010          WORK = A(I,J)
      000014          A(I,J) = A(J,I)
      000021        1 A(J,I) = WORK
      000027          RETURN
      000027          END  MATFLIP
```

```
                      SUBROUTINE MATI GEN(A,EVALUES,EV CTORS,N)                          68.0209
                *     SOURCE DATE 68.0209        CONVERT TO CDC 6400
                C     SOURCE DATE 67.0728   BRAND NEW CODE  BY PERKINS
                C
                C     MATEIGEN FINDS THE EIGEN VALUES AND UNIT EIGEN VECTORS OF THE
                C     N X N MATRIX A.  THE EIGEN VALUES ARE RETURNED IN EVALUES IN THE
                C     ORDER MAX TO MIN.  THE UNIT EIGEN VECTORS ARE COLUMNS OF THE
                C     EVECTORS MATRIX.  THE FIRST COLUMN IS THE VECTOR FOR THE FIRST
                C     EIGEN VALUE ETC.  SEE JACOBI FOR CALCULATION DETAILS.
                C
000006                DIMENSION A(N,N),EVALUES(N),EV CTORS(N,N)                        68.0209
                C
                C     SAVE DIAGONAL ELEMENTS IN EVALUES
000006                DO 1 I=1,N
000007                EVALUES(I)=A(I,I)
000017              1 CONTINUE
                C
                C     CALCULATE EIGEN VALUES
000021                CALL JACOBI(A,EV CTORS,N,1,E-10,1,IT)                            68.0209
                C
                C     SWITCH DIAGONAL TERMS AND EIGEN VALUES
000024                DO 2 I=1,N
000030                WORK=EVALUES(I)
000032                EVALUES(I)=A(I,I)
000036                A(I,I)=WORK
000040              2 CONTINUE
                C
                C     RESTORE OFF DIAGONAL TERMS TO A UPPER RIGHT TRIANGLE IS UNCHANGED
                C     BY JACOBI
000042                M=N-1
000043                DO 3 I=1,M
000045                K=I+1
000047                DO 3 J=1,I
000050                A(K,J)=A(J,K)
0000A1              3 CONTINUE
000066                RETURN
000067                END  MATIGEN
```

PRECEDING PAGE BLANK-NOT FILMED

```
                      SUBROUTINE MATINV(A,N,NMAX,B,M,PIVOT,IPIVOT,INDEX,DETERM)          00000000
                *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                C     FI COUA MATINV  MATRIX INVERSION                                   00000010
                C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS    00000020
                C                                                                        00000040
                C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS    00000020
                C                                                                        00000040
000012                DIMENSION IPIVOT(N),A(NMAX,NMAX),B(NMAX,1),INDEX(N,2),PIVOT(N)     00000050
                C                                                                        00000060
                C     INITIALIZATION                                                     00000070
                C                                                                        00000080
000012                DETERM=1.0
000014                DO 20 J=1,N
000016             20 IPIVOT(J)=0                                                        00000110
000022                DO 550 I=1,N
                C                                                                        00000130
                C     SEARCH FOR PIVOT ELEMENT                                           00000140
                C                                                                        00000150
000023                AMAX=0.0
000024                DO 105 J=1,N
000026                IF (IPIVOT(J)-1) 60, 105, 60
000031             60 DO 100 K=1,N                                                       00000190
000033                IF (IPIVOT(K)-1) 80, 100, 740
000036             80 IF (ABSF(AMAX)-ABSF(A(J,K))) 85, 100, 100                         00000210
000046             85 IROW=J                                                             00000220
000050                ICOLUM=K
000051                AMAX=A(J,K)
000055            100 CONTINUE                                                          00000250
000060            105 CONTINUE                                                          00000260
000063                IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
                C                                                                        00000280
                C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                  00000290
                C                                                                        00000300
000066                IF (IROW-ICOLUM) 140, 260, 140
000070            140 DETERM=-DETERM                                                    00000320
000072                DO 200 L=1,N
000073                SWAP=A(IROW,L)
000077                A(IROW,L)=A(ICOLUM,L)
000104            200 A(ICOLUM,L)=SWAP                                                  00000360
000110                IF(M) 260, 260, 210
000111            210 DO 250 L=1,M                                                      00000380
000113                SWAP=B(IROW,L)
000117                B(IROW,L)=B(ICOLUM,L)
000124            250 B(ICOLUM,L)=SWAP                                                  00000410
000132            260 INDEX(I,1)=IROW                                                   00000420
000136                INDEX(I,2)=ICOLUM
000141                PIVOT(I)=A(ICOLUM,ICOLUM)
000146                DETERM=DETERM*PIVOT(I)
                C                                                                        00000440
                C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                                  00000470
                C                                                                        00000480
000151                A(ICOLUM,ICOLUM)=1.0
000153                DO 350 L=1,N
000155            350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
000170                IF(M) 380, 380, 360
000171            360 DO 370 L=1,M                                                      00000540
000177            370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
                C                                                                        00000550
                C     REDUCE NON-PIVOT ROWS                                              00000560
                C                                                                        00000570
000216            380 DO 550 L1=1,N                                                     00000580
000220                IF(L1-ICOLUM) 400, 550, 400
000222            400 T=A(L1,ICOLUM)                                                    00000600
000233                A(L1,ICOLUM)=0.0
000234                DO 450 L=1,N
000236            450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                     00000630
000253                IF(M) 550, 550, 460
```

PRECEDING PAGE BLANK-NOT FILMED

```
000254      460 DO 500 L=1,M                                                00000650
000256      500 A(I1,L)=B(I1,L)-B(ICOLUM,L)*T                               00000660
000273      550 CONTINUE                                                    00000670
                                                                            00000680
            C                                                               00000690
            C     INTERCHANGE COLUMNS                                       00000700
            C
000300            DO 710 I=1,N
000301            L=N+1-I
000303            IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                  00000740
000314      630   JROW=INDEX(L,1)
000322            JCOLUM=INDEX(L,2)
000326            DO 705 K=1,N
000327            SWAP=A(K,JROW)
000334            A(K,JROW)=A(K,JCOLUM)
000341            A(K,JCOLUM)=SWAP                                          00000800
000343      705   CONTINUE                                                  00000810
000344      710   CONTINUE                                                  00000820
000347      740   RETURN
000350            END  MATINV
```

```
                SUBROUTINE MATMULT (A,B,C,NRA,NCARB,NCB)
          C     SOURCE DATE 67.0726   HHAND NEW CODE
          C
          C      PERKINS, C. W.   GENERAL REAL MATRIX MULTIPLICATION
          C      C = A * B, WHERE A,B,AND C ARE MATRICES
          C      NRA = NUMBER OF ROWS IN A (AND C)
          C      NCARB = NUMBER OF COLUMNS IN A AND ROWS IN B
          C      NCB = NUMBER OF COLUMNS IN B (AND C)
          C
          C      *** NOTE *** MATRIX C MUST NOT OCCUPY THE SAME STORAGE AS
          C      MATRICES A OR B IN THE CALLING SEQUENCE.
          C
000007            DIMENSION A(NRA,NCARB), B(NCARB, NCB), C(NRA,NCB)
          C
000007            DO 1  I=1,NRA
000011            DO 1  J=1,NCB
000012            SUM = 0.
000013            DO 2  K=1,NCARB
000015            SUM = SUM + A(I,K) * B(K,J)
000030          2 CONTINUE
000032            C (I,J) = SUM
000036          1 CONTINUE
000042            RETURN
000042            END  MATMULT
```

```
                SUBROUTINE MAT NVRT(A,N)                                        68.0209
            C   SOURCE DATE 68.0220   CORRECT CALLING SEQ TO MATINV
            *   SOURCE DATE 68.0209       CONVERT TO CDC 6400
            C   SOURCE DATE 67.0728   BRAND NEW CODE  BY PERKINS
            C
            C   MATINVRT RETURNS A=INVERSE WHERE A IS AN N X N MATRIX,
            C**** NOTE N IS LIMITED TO 24.  IF DESIRED N EXCEEDS THIS LIMIT, SEE MATINV.
            C
000004          DIMENSION  A(N,N),WORK(96)
            C
000004          CALL MATINV(A,N,N,B,0,WORK,WORK(25),WORK(49),0)                 68.0220
000017          RETURN
000020          END  MATNVRT
```

```
                SUBROUTINE  MATPAN P   (A,B,NRA,NCA,KODER,KODEC)                    68.0209
              *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
              C     SOURCE DATE 67.0728   BRAND NEW CODE  BY PERKINS
              C
              C     MAT P AND P PERMUTES AND/OR PARTITIONS MATRICES
              C
              C     A IS AN NRA X NCA MATRIX
              C     B IS A SUB MATRIX OF A, CHOSEN ACCORDING TO KODER AND KODEC FOR
              C     ROWS AND COLUMNS RESPECTIVELY.
              C     KODER AND KODEC ARE STRINGS OF DIGITS CORRESPONDING TO THE ROWS
              C     AND COLUMNS OF A WHICH ARE TO BE TRANSMITTED TO B.  THE NUMBER OF
              C     ROWS AND COLUMNS OF B MUST CORRESPOND TO THE NUMBER OF DIGITS IN
              C     KODER AND KODEC.  IF KODEC IS ZERO,KODER IS USED FOR COLUMNS ALSO.
              C
              C     EXAMPLE, NRA=6,NCA=5,KODER=315,KODEC=523
              C     B IS THEREFORE A 2 X 3 ARRAY AND
              C     B(1,1)=A(3,5)
              C     B(1,2)=A(3,2)
              C     ETC.
              C
              C *** NOTE, A AND B ARE LIMITED TO A MAXIMUM OF 9 X 9
              C
000007                DIMENSION A(NRA,NCA),B(2),KR(9),KC(9)
              C
              C
              C     DECODE KODER AND KODEC
000007                KDR=KODER
000010                KDC=KODEC
000011                DO 1 I=1,9
000013                KR(I)=MOD(KDR,10)
000020                IF(KR(I).EQ.0) GO TO 2
000022                IF(KR(I).GT.NRA) GO TO 90
000025               11 KDR=KDR/10
000030                1 CONTINUE
000032                2 NRB=I-1
000034                  IF(KDC.EQ.0) GO TO 5
000035                  DO 3 I=1,9
000037                  KC(I)=MOD(KDC,10)
000044                  IF(KC(I).EQ.0) GO TO 4
000046                  IF(KC(I).GT.NCA) GO TO 90
000051               12 KDC=KDC/10
000054                3 CONTINUE
000056                4 NCB=I-1
000060                  GO TO 6
              C       IF KODEC IS ZERO KODER IS USED
000061                5 CALL XMIT(NRB,KR,KC)
000064                  NCB=NRB
000066                A CONTINUE
              C
              C     OPERATE ON B MATRIX
000066                DO 7 I=1,NRB
000073                NIA=NRB-I+1
000075                IA=KR(NIA)
000100                DO 7 J=1,NCB
000101                K=J+NRB*(I-1)
000105                NJA=NCB-J+1
000107                JA=KC(NJA)
000111                B(K)=A(IA,JA)
000117                7 CONTINUE
000123                RETURN
              C
000124               90 CALL TRA DEBR(3)MCALLING SEQ ERROR IN MATPANP - )          68.0209
000126                RETURN
000127                END  MATPANP
```

PRECEDING PAGE BLANK-NOT FILMED

```
                SUBROUTINE MATRANS  (A, B, NRA, NCA)                                68.0209
          *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
          C     SOURCE DATE 67.0726   BRAND NEW CODE
          C
          C      PERKINS, C. W.   MATRIX TRANSPOSE
          C      A = AN NRA BY NCA MATRIX
          C      B = TRANSPOSE OF A
          C      NRA = NUMBER OF ROWS IN A
          C      NCA = NUMBER OF COLUMNS IN A
          C
          C      *** NOTE - IN CALL STATEMENT, DO NOT USE SAME LOCATION
          C                 FOR BOTH FIRST AND SECOND ARGUMENTS
          C
000006          DIMENSION A(NRA,NCA), B(NCA,NRA)
          C
000006          DO 1  I=1,NRA
000007          DO 1  J=1,NCA
000010          B(J,I)=A(I,J)
000021        1 CONTINUE
000026          RETURN
000026          END  MATRANS
```

```
                      SUBROUTINE MATSCAL  (X,A,NR,NC)                                   68.0209
                C     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                C     SOURCE DATE 67.0726   BRAND NEW CODE
                C
                C         MULTIPLIES MATRIX A (WHICH IS NR ROWS BY NC COLUMNS) BY X
                C
000006                DIMENSION A(NR,NC)
                C
000006                DO 2 K=1,NC
000007                DO 2 L=1,NR
000010           2    A(L,K)=X*A(L,K)
000022                RETURN
000022                END  MATSCAL
```

```
C     SUBROUTINE MINMIZE (YNEW,XNEW,WORK)
C     CALL SETVAL(IR,3,ZERO,WORK)    INITIALIZE TO ZEROS
C     CALL SETC (IR,3,1A)    BREAKPOINT = 3
C
C     YNEW IS DUMMY, TO MINIMIZE YNEW,.. MINIMIZE RETURNS A FLAG
C     TO INDICATE PROGRESS OF CONVERGENCE.
C
C     MINMIZE MUST BE INITIALIZED AS FOLLOWS .....
C     CALL SETVAL (DX,PRCSN,WORK       WHERE ....
C                  DX IS INITIAL INCREMENT TO BE USED ON XNEW
C                  PRCSN IS ALLOWABLE TOLERANCE ON XNEW
C                  WORK IS A-WORD WORK-SPACE USED BY MINMIZE
C
C     MINMIZE RETURNS FUNCTION VALUE =  0 IF CONVERGENCE PROCEEDING NORMALLY
C                                      +1 IF CONVERGENCE COMPLETED
C                                      -1 IF FAILURE TO CONVERGE
C
C     VARIABLE DEFS AS FOLLOWS .....
C       DIMENSION WORK(8)
C       CALL SETVAL (10,3,ZERO,WORK)
C       EQUIV
C     1 TEST=F(1)
C       IF MINMIZE (TEST,X,WORK) 20,10,30
C     2   ERROR MESSAGE
C     10  PROCEED
C
000005        DIMENSION WORK(8),WORK(8),X(2),Y(2)
000005        EQUIVALENCE (WORK(1),DX),(WORK(2),PRCSN),(WORK(3),XPHASE)
000005        EQUIVALENCE (WORK(4),XOLD),(WORK(5),YOLD),(WORK(7),Y)
000005        DATA ZERO/0.0/
C
000005        CALL EXIT(8,WORK,WORK)
000007        XPHASE=XOLD
000011        GO TO(
000012        IF (    ),(    ),(    ),XPHASE
000016        IF (    ),(    ),(    ),XPHASE
C             XNEW=XOLD+DX
000021        GO TO (
000022     1  DX=
000035        DATA A1/    /,A2/    /,A3/    /,A4/    /,A5/    /,A6/    /
000043        FORMAT(6E12.5)
C             * ITERATION #13/
C          1  6E13.5,13(/12X,6E13.5))
C             XNEW=X(1)+DX
000074        CALL EXIT(8,XNEW,WORK)
000076        CALL PRINT(6HPRCHAIN,'IF THIS PROGRAM =',
C                  2,3HIF
C
000111        GO TO(    20,2A,30,7A),XPHASE
000121     22 XPHASE=2
000122     2A XNEW=X+DX
C             IF(ABS(DX)-PRCSN) MINMIZE=1
C             DX=-DX/2.
000131        IF(XNEW-XOLD) 90,90,2A
000133     2A XPHASE=3
000134        IF (Y(1)-Y(2)) 90,90,2A
C             XNEW=
000141        Y(1)=Y(2)
000142        XOLD=X(2),YOLD
000143        GO TO 90
000144     30 IF(YNEW-Y(2)) 90,90,2A
000147        XPHASE=4
000150        DX=(X(1)-X(2))*(Y(2)-YNEW)/(2.0*(Y(1)-2.0*Y(2)+YNEW)) + DX
000156        XNEW=X(2)+DX
000157        IF(ABS(DX)-PRCSN) MINMIZE=1
000163        GO TO 90
000166     7A XPHASE=3   *   GO TO 2A
C
000166     90 CALL EXIT (8,WORK,WORK)
000170        RETURN
```

PRECEDING PAGE BLANK-NOT FILMED

```
000172                  ;ALPHA SEQUENCE                                 68.0816
000211                  ;ALL UNIT NUMBERS
000204                  ;ACCUMULATOR
000207                  ;ALPHABETIC
000210                  ;KEYLOG
000211                  ;UNIT NO
000212                  END OF SEQUENCE
```

PRECEDING PAGE BLANK-NOT FILMED

```
                      * [illegible] MESSAGE)
                      *  [illegible]          CONVERT TO CDC 6400
                      *  [illegible]          CHANGE NEW CODE
                      *     [illegible] INTRA-CHAIN ERROR CONDITIONS
                      *
                      *     [illegible] WRITES AN ERROR-TRACE MESSAGE AND EXITS.
                      *     [illegible] SWITCH IFERR HAS BEEN SET NON-ZERO. HOWEVER,
                      *     [illegible] WILL BE RETURNED TO YOUR PROGRAM. (IT IS THEN YOUR
                      *     [illegible] RESPONSIBILITY TO CONSULT THE ERROR-COUNTER NUMERR
                      *     [illegible] ERROR CONDITIONS)
                      *
                      *     [illegible] NUMBER OF ERRORS WHICH HAVE OCCURRED.
                      *     [illegible] WAY
                      *     [illegible] THE NUMBER OF ERRORS.
                      *     [illegible]
                      *     [illegible] THE ERROR-CONTROL SWITCH.
                      *     [illegible]
                      *
000000                DATA IFERR=0,NUMERR=0
                      *
000000                ENTRY [illegible]
000000                RETURN
                      *
000006                ENTRY TRACERR                                    68.0209
000013                CALL TITLERR(2,MESSAGE)
00001A                CALL AR ERROR(1,2H=CHAIN OF CALLING PROGRAMS =.)  68.0209
000020                NUMERR=NUMERR+1
000022                IF (IFERR .NE. 0) RETURN
000024                CALL REALEX
00002A                CALL EXIT
                      *
000027                ENTRY SETIFER                                    68.0209
000035                IFERR=MESSAGE
000037                RETURN
                      *
                      ENTRY SETNERR                                    68.0209
                      NUMERR=MESSAGE
000050                RETURN
                      *
                      END MISTAKE
```

```
                      REAL FUNCTION MIXER(X)                                    1367
                C     SOURCE DATE 66.0101   BRAND NEW CODE                      1368
                C                                                               1369
                C         RETURNS X WITH AN INTEGER NAME                        1370
                C                                                               1371
000003                    MIXER=X                                               1372
000004                    RETURN
000005                    END   MIXER
```

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE OLDATA (NAME)
C     SOURCE DATE A7.0720   BRAND NEW CODE
C
C     WRITES AND CHANGE= MESSAGE FOR S PROGRAM =NAME=
C
      COMPX$,RASCON
000003      COMMON /RASCON/
     1   PROGRM,   #PAGE,    LINE,     TODAY,    RUN,      PRINT(A),
     2   MSG,      FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
     3   KLASS,    KGROUP,   KUNITS,   KSTINT,   KOORD,    TREND
     4  ,NOFILE
C
C
000003      LINE=LINE+1
000005      WRITE(6,1) NAME
000012    1 FORMAT(20X,*SUBR *A8,* -  NO CHANGE IN DATA*)
000012      RETURN
000013      END  OLDATA
```

```
      SUBROUTINE TIME (DPRI,STATE,VRHS )                                    1376
*     SOURCE DATE AA.0204        CONVERT TO CDC 6400
*     SOURCE DATE AA.0806    ALLOW HYPERBOLIC ORBITS
*     SOURCE DATE AA.1307
C                                                                           1378
C         VECTORS ORBITAL ELEMENTS CORRESPONDING TO GIVEN STATE VECT        1379
C                                                                           1380
C         ELLIPTIC OR HYPERBOLIC ORBITS                                     1381
C                                                                           1382
C         STATE  -  STATE VECTOR (ACCELERATION COORDINATES IN VRHS)         1383
C         DPRI   -  ORBITAL ELEMENT VECTOR                                  1384
C                                                                           1385
C         DPRI(1)=1/A                                                       1386
C                                                                           1387
      COMMON/CONCON/
000000*       1   P ,       SMU,       SLV,       SMF,       SKP,       OMEGA,
               2   OMEGA,     GCON,      OMEGP,     RHO2R2,    TWOPI,     HALFPI
C
000004        DIMENSION STATE(10),DPRI(10),S(10),AXES(10,3),DE(10)          1396
000004        EQUIVALENCE (S(1),T),(S(2),R),(S(5),VR),(S(6),VU)             1397
000004        EQUIVALENCE (DE(1),GAMMA),(DE(5),P),(DE(6),E),(DE(7),TP),(DE(8),
              1 PP),(DE(9),A)                                               1398
000004        DATA /AXES=30(0)/
C
000004        DO 2 I=1,10                                                   1400
000005    2   S(I)=STATE(I)                                                 1401
000012        CALL UNITV(S(2),AXES(2,1))                                    1402
000015        CALL CROSS(S(2),S(5),AXES(2,3))                               1403
000020        CALL CROSS(AXES(2,3),AXES(2,1),AXES(2,2))                     1404
000025        CALL EULANG(1,AXES,DE,0)                                      1405
000030        CALL TRNSFM(S,AXES,S,1,1)                                     1406
000034        DE(1)=T                                                       1407
000036        A=GCON*FDIV(P/(2.-GCON-R*(VR**2+.0**2)))                      AA.0209
000047        P=(R*VU)**2/GCON                                              1409
000051        ES=SQRT(1.-(VR*A-P)/A)                                        AA.0210
000061        ANOM=ACOS((FDIV(P-R)/(E*R)))                                  AA.0211
000072        IF (VR.LT.0.) OMEGP=-0                                        1412
000075        ARA=ABS(A)                                                    AA.0212
000107        TP=ANA*SQRT(ARA/GCON)                                         AA.0213
000120        TP=T-SQRT(1.0/AXES(2))                                        1414
000111        GAMMA=ATAN(                                                   1415
000113        CALL XMIT(10,DE,DPRI)                                         AA.0214
000120        RETURN                                                        1417
000121        END   SUBR
```

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE ORB2(MODE,VALUE,S1,S2,ORBEL)                                1414
C     SOURCE DATE AR,7 74   ALLOW ZERO  VELOCITY ELEVATION AT S1   AA,0002
C     SOURCE DATE AR,0274     CONVERT TO CDC AA00
C     SOURCE DATE AR,0714   CALL TRAJCHK IF ERROR CONDITION
C     SOURCE DATE AA,0406  ALLOW HYPERBOLIC ORBITS
C     SOURCE DATE AA,1279
C                                                                            1411
C        DETERMINES ELEMENTS OF KEPLER ORBIT JOINING TWO POINTS              1412
C                                                                            1413
C        ELLIPTIC OR HYPERBOLIC ORBITS                                  AA,0004
C                                                                       AA,0005
C        MODE   -  SELECTS FINAL DETERMINANT OF ORBIT                        1414
C        VALUE  -  VALUE OF FINAL DETERMINANT OF ORBIT                       1415
C        S1     -  INITIAL STATE ON ORBIT (POSITION AND TIME)                1416
C        S2     -  FINAL STATE ON ORBIT (POSITION ONLY)                      1417
C        ORBEL  -  ORBITAL ELEMENT VECTOR RETURNED BY ORB2                   1418
C                                                                            1419
C        THE FOLLOWING ORBIT DETERMINATIONS ARE SELECTABLE -                 1420
C                                                                            1421
C        MODE       MEANING OF #VALUE#                                       1422
C        ----       ----------------------------------------------------     1423
C          1        FLIGHT-TIME FROM S1 TO S2
C          2        EXCESS FLIGHT-TIME (IE, DESIRED-FLIGHT-TIME
C                                  MINUS  MINIMUM-ENERGY-FLIGHT-TIME)
C          3        SPEED AT S1.  FOR ELLIPTIC ORBITS, NEGATIVE VALUE   AA,0004
C                   RETURNS ORBIT FOR MINIMUM-TIME FLIGHT AND POSITIVE  AA,0005
C                   VALUE RETURNS MAXIMUM-TIME ORBIT.                   AA,0006
C          4        LOCAL VELOCITY ELEVATION AT S1                           1429
C                                                                            1431
C        IF VALUE EQUALS ZERO FOR MODES OF 1 OR 2, A MINIMUM-ENERGY          1432
C        ELLIPTIC TRAJECTORY WILL BE RETURNED                           AA,0004
C                                                                            1436
C        ORB2 ENTERS THE ARRIVAL TIME AT S2 IN S2(1).                        1435
C                                                                            1436
C        WRITTEN 12/9/66                                                     1437
C                                                                            1438
      COMPOOL,C3POOL
      COMMON /CCC-COM/
     1   PI,        SRD,       SIX,       SME,       SKD,       ...
     2   DATE,      CCC,       WHOAT,     WHO2PO,    INOUT,     ...
C
      DIMENSION S1(1),S2(1),ORBEL(10),AXES(3,3),...
      EQUIVALENCE (DIS,XP),(VIN,YP),(ZIN,ZP),(DTM,WP),...
      DATA (AXES=0.0)
C
      MODE=IABS(MODE)                                                        1441
      PI=XMAG(S1(2))                                                         1444
      P2=XMAG(S2(2))                                                         1445
      THETA=SEPA(S1(2),S2(2))                                                1446
      CALL UNITV(S1(2),AXES(2,1))                                            1447
      CALL CROSS(S1(2),S2(2),AXES(2,3))
      CALL CROSS(AXES(2,3),AXES(2,1),AXES(2,2))                              1450
      CALL FILANG(1.,AXES,0.,3)                                              1451
      C(1)=S1(1)                                                             1452
      DO 3 I=2,4                                                             1453
    1 CHORD(I)=S2(I)-S1(I)                                                   1454
      C=XMAG(CHORD(2))                                                       1456
      S=(P1+P2+C)/2.                                                         1457
      IF (MODE.GT.2) GO TO 30                                           AA,0006
C                                                                       AA,0007
C     ELLIPTIC OR HYPERBOLIC ORBITS -- MODES 1 AND 2                    AA,0008
C                                                                       AA,0009
      TRAP=SQRT(2./GCON)*(SQRT(S**3)-SQRT((S-C)**3))/3.                 AA,0010
      A=S/2.                                                                 1458
      PP=A*SQRT(A/GCON)                                                      1459
      X=PI                                                                   1460
      Y=2.*ASIN(SQRT((S-C)/S))                                               1461
      T12MIN=PP*(P(-X+SIN(X))                                                1462
```

```
000142          TCOMP=.A  F                                                     AA.  0QA
000146          IF (TCOMP .EQ. 2.) GO TO 9                                      AA.  0QA
000147          IF (MODE .EQ. 2) TCOMP=TCOMP-T12MIN                             AA.  0QA
000173          IF (TCOMP .LE. TRMHL) GO TO 100                                 AA.  0QA
                C                                                               AA.  0QA
                C     ELLIPTIC ORBITS -- MODES 1 AND 2                          AA.  0QA
                C                                                               AA.  0QA
000176        9 DDAL=T12MIN                                                     AA.  0QA
000200          ANO=T12MIN                                                          1*A4
000201          AMN=A                                                               1*A5
000202          IF (VALUE.EQ.0.) GO TO 25                                           1*A6
000203          ERR=.                                                         AA.  0*14
000205          DDAL=VALUE                                                          1*A8
000206          IF (MODE.EQ.2) G AL=T12MIN-DDAL                                     1*A9
000212          GO=1                                                          AA.  0QA
000213          NTRY=1                                                              1*B2
000214       10 NTRY=NTRY+1                                                         1*B7
000216          PP=A*SQRT(A/GCON)                                                   1*B2
000223          XA=2.*ASIN(SQRT((S-A)/(2.*A)))                                      1*B3
000233          XB=2.*ASIN(SQRT((S-C)/(2.*A)))                                      1*B4
000244          OLAST=ANO                                                           1*B5
000245          IF (DDAL.GT.T12MIN) GO TO 14                                        1*B6
000256          ANO=PP*(XA-SIN(XA)-XB+SIN(XB))                                      1*B7
000264          GO TO 15                                                            1*B8
000270       14 ANO=PP*(TWOPI-XA+SIN(XA)-XB+SIN(XB))
000312       15 IF (ABS(ANO-DDAL).LE.PP*.OKERR) GO TO 25                            *A
000313          IF (NTRY.GE.20) GO TO 20                                            1*C2
000317          DA=(DDAL-ANO)*(A-OLAST)/(ANO-OLAST)                                 1*C3
000326          IF (NTRY.EQ.10) DA=A/100.                                           1*C4
000326          IF (DA.LT.-AMN) DA=.7*(AMN-A)                                       1*C5
000332          A=A+DA                                                              1*C6
000334          GO TO 10                                                            1*C7
                C
000345       20 CALL ERAD(15)                                                       1*C8
000347          CALL STOPT &HSUBROUTINE KEPH HAS FAILED TO CONVERGE AFTER 20 ITERA
                1 TIONS.,X,5)                                                       1*C9
000364          CALL STORT(A,111,XA,52,-1)                                          1*D1
000376          IF(JOG.NE.2) WRITE(6,222) MODE,VAL,A,T12MIN,DDAL,XA,XB,ANO,PP,S,C
000413       22 FORMAT(/&MODE     VALUE     T12MIN      DDAL                      AA.  0QA
                * A           DA*/1A,5E10.3,4*/10.2)                            AA.  0QA
000431          CALL TRA ERR(IN )                                               AA.  0QA
000432          RETURN                                                          AA.  0QA
                C
000433       25 X=XA+TH                                                             1*D2
000435          IF (DDAL.GT.T12MIN) X=XA-TH                                         1*D3
000421          P=A*A*(S-A)*(S-B)*SIN(XA/2.)/2.)**2                                 1*D4
000435          E=SQRT(ECNV/(A-P)/A))                                         AA.  0QA
000444          D1=ACOS(ECNV*(P-A)/(A*E))                                     AA.  0QA
000450          IF (ABS(X/(1.+E*COS(D1+DEL TA))-R2)/R2.GT..0000011 GO TO 1
000471          S2(1)=X(1)+GNO                                                      1*D5
000477          GO TO 90                                                            1*D6
                C                                                               AA.  0QA
                C     HYPERBOLIC ORBITS -- MODES 1 AND 2                        AA.  0QA
                C                                                               AA.  0QA
000477       100 DDAL=TCOMP                                                     AA.  0QA
000501          S2(1)=DDAL+X(1)                                                 AA.  0QA
000503          BB=SQRT(S)                                                      AA.  0QA
000505          CC=SQRT(S-C)                                                    AA.  0QA
000511          JOG=2                                                           AA.  0QA
000512          NTRY=1                                                          AA.  0QA
000513          A1=A,*GCON*DDAL**2/C**2                                         AA.  0QA
000517       110 NTRY=NTRY.1                                                    AA.  0QA
000521          IF (NTRY .EQ. 30) GO TO 20                                      AA.  0QA
000526          DD=SQRT(S+2.*A1)                                                AA.  0QA
000533          EE=SQRT(S-C+2.*A1)                                              AA.  0QA
000541          A=GCON*DDAL**2/(BB*DD-CC*EE+2.*A1*ALOG((CC+EE)/(BB+DD)))**2     AA.  0QA
000557          ERROR=ABS(A1-A)/A1                                              AA.  0QA
000562          IF (ERROR .LE. 0.000001) GO TO 125                              AA.  0QA
000567          A1=A                                                            AA.  0QA
000570          GO TO 110                                                       AA.  0QA
                C                                                               AA.  0QA
000571       30 IF (MODE.GE.4) GO TO 60                                             1502
                C                                                               AA.  0QA
                C     ELLIPTIC OR HYPERBOLIC ORBITS -- MODE 3                   AA.  0QA
                C                                                               AA.  0QA
000574          V=VALUE                                                             1503
000575          A=GCON*R1/(2.*GCON-R1*V**2)                                         1505
000601          IF (A .LT. 0.) GO TO 125                                        AA.  0QA
000603          U=DELTA/2.                                                          1504
                C                                                               AA.  0QA
                C     ELLIPTIC ORBITS -- MODE 3                                 AA.  0QA
                C                                                               AA.  0QA
```

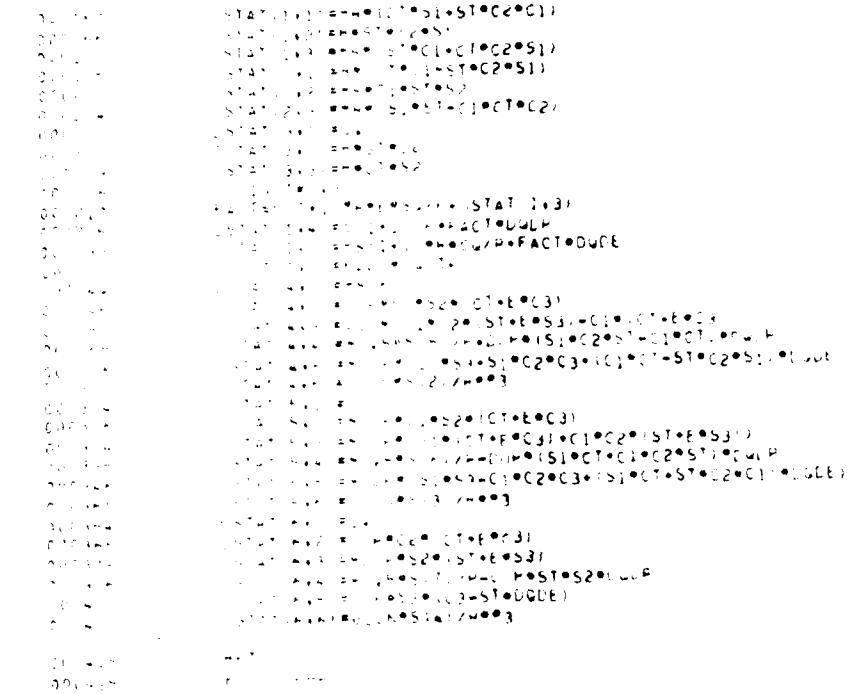
```
000605          SNU=SIN(U)                                                      1504
000607          CSU=COS(U)                                                      1507
000611          C1= (((R1+R2)/CSU)**2+((R1-R2)/SNU)**2)*(A/(R1*R2))**2          1508
000623          C2= 4.-4.*A*(R1+R2)/(R1*R2*CSU**2)                              1509
000632          C3=-4.+4./CSU**2                                                1510
000635          RADICL=C2**2-4.*C1*C3                                           1511
000660          IF (RADICL.LT.0.) CALL TITLER(-1,7HORBR2 FINDS SPECIFIED VELOCITY 1512
                1INADEQUATE TO REACH DESIRED GOAL * * * - )                     1513
000667          RADICL=SQRT(AMAX1(0.,RADICL))                                   1514
000676          E=SQRT(AMAX1(0.,1.+.5*(C2/C1+SIGN(RADICL/C1,V))))               1515
000774          P=A*(1.-E**2)                                                   1516
000677          U=ATAN2(P*(1./R2-1./R1)/SNU,(P*(1./R2+1./R1)-2.)/CSU)           1517
000716          IF (U.GT.0.) U=U-TWOPI                                          1519
000721          U1=-U-U                                                         1520
000724    39    PP=A*SQRT(A/BCON)                                               1521
000732          GO TO 90                                                        AA,0A04
                C                                                               AA,0A04
                C     HYPERBOLIC ORBITS -- MODES 1,2, AND 3                     AA,0A04
                C                                                               AA,0A04
000735    125   A=ABS(A)                                                        AA,0A04
000737          GAM=2.*ALOG(SQRT(.5*S/A)+SQRT(.5*S/A+1.))                       AA,0A04
000757          DEL=2.*ALOG(SQRT(.5*(S-C)/A)+SQRT(.5*(S-C)/A+1.))               AA,0A04
001001          HFBD=(GAM+DEL)/2.                                               AA,0A04
001004          P=A*(S-R1)*(S-R2)*(EXP(HFBD)-EXP(-HFBD))**2/C**2                AA,0A04
001021          E=SQRT((A+P)/A)                                                 AA,0A04
001025          PP=A*SQRT(A/BCON)                                               AA,0A04
001032          U1=ACOS((P-R1)/(R1*E))                                          AA,0A04
001041          IF (ABS(P/(1.+E*COS(U1+DELTA))-R2)/R2 .GT. 0.000001) U1=TWOPI-U1 AA,0A04
001056          GO TO 90                                                        AA,0A04
                C                                                               AA,0A04
                C     ELLIPTIC OR HYPERBOLIC ORBITS -- MODE 4                   AA,0A04
                C                                                               AA,0A04
001062    40    TNG=FDIV(SIN(VALUE)/COS(VALUE))                                 AA,0209
001106          U1=ATAN((R2*TNG*COS(DELTA)-1.)/(R1-R2+R2*TNG*SIN(DELTA)))       1523
001124          IF (TNG.GE.0..AND.U1.LT.0..OR.TNG.LE.0..AND.U1.GT.0.) U1=U1+PI  1524
001145          IF(TNG.NE.0)GO TO 45                                            AA,7024
001151          DIV=R2*(1.+COS(DELTA))-2.*R1                                    AA,7024
001160          IF(R1-R2)60,45,70                                               AA,7024
                C     S1 AT PERIGEE OR ORBIT HYPERBOLIC                         AA,7024
                C     COMPARE S2 WITH CORRESPONDING POSITION ON  THE PARABOLA   AA,7024
001165    60    Y2=DOT(S2(2),S1(2))/R1                                          AA,7024
001201          X2=SQRT(R2**2-Y2**2)                                            AA,7024
001206          Y2=R1-Y2                                                        AA,7024
001210          IF(Y2.GT.X2**2/(4.*R1))GO.65                                    AA,7024
                C     HYPERBOLIC                                                AA,7024
001222    65    A=ABS(R1*(-1.+(R2-R1)/DIV)) $ E=1.+R1/ABS(A)                    AA,7024
001233          GO TO 67                                                        AA,7024
001234    66    A=R1*(1.+(R1-R2)/DIV) $ E=1.-R1/A                               AA,7024
001243          GO TO 67                                                        AA,7024
001244    70    A=R1*(1.+(R2-R1)/DIV)                                           AA,7024
001251          E=R1/A-1.                                                       AA,7024
001253          U1=PI                                                           AA,7024
001254          GO TO 67                                                        AA,7024
001255    45    P=FDIV(TNG/(SIN(U1)-COS(U1)*TNG))                               AA,0209
001267          A=R1*(1.+E*COS(U1))/(1.-E**2)                                   1526
001277    67    P=A*(1.-E**2)                                                   1527
001303          A=ABS(A)                                                        AA,0A04
001304          PP=A*SQRT(A/BCON)                                               AA,0A04
                C                                                               AA,0A04
                C     ELLIPTIC OR HYPERBOLIC ORBITS -- ALL MODES                AA,0A04
                C                                                               AA,0A04
001311    90    O(4)=O(4)-U1                                                    AA,0A04
001313          O(4)=AMOD(O(4),TWOPI)  $  IF(O(4).LT.0.) O(4)=O(4)+TWOPI        AA,0A04
001321          IF(E.GT.1.)A=-A                                                 AA,7024
001325          TP=O(1)-ORBTIM(0.,U1,A,E,PP)                                    1530
001333          IF (MODE.GE.3) S2(1)=ORBTIM(0.,U1,DELTA,A,E,PP)-TP              1531
001353          CALL XMIT(19,0,ORBEL)                                           AA,0A04
001356          RETURN                                                          1534
001357          END  ORBR2
```

```
      SUBR[illegible] (MODE,VALU,ORBEL,STATE)
C     [illegible]   REMOVE CALL OF RITEF
C     [illegible]   REVISE TEST FOR APOGEE/PERIGEE
C     [illegible]   CALL RITEF, NOT ORIO / CHECK FOR ILLEGAL RADIUS
C     [illegible]   CONVERT TO CDC ANOT
C     [illegible]   CORRECT POTENTIAL OVERFLOW ERROR
C     [illegible]   ADD OPTIONAL MODES 4,5
C     [illegible]   CALL TRAIDERR IF ERROR CONDITION
C     [illegible]   USE ECC ANOMALY AS ITERATION VARIABLE
C     [illegible]   USE RADIUS/ANGLE/TIME AS ITERATION VARIABLE
C                                                                       1537
C        RETURNS STATE VECTOR OF A POINT ON A KEPLER ORBIT              1538
C                                                                       1539
C        MODE   -  SELECTS SPECIFICATION OF POINT IN ORBIT              1540
C                  0  -  VALUE IS TRUE ANOMALY                          1541
C                  1  -  VALUE IS RADIUS (RADIUS INCREASING IN TIME)    1542
C                  2  -  VALUE IS RADIUS (RADIUS DECREASING IN TIME)    1543
C                  3  -  VALUE IS TIME AT WHICH POINT IS REACHED        1544
C                  4  -  VALUE IS ALTITUDE (INCREASING IN TIME)
C                  5  -  VALUE IS ALTITUDE (DECREASING IN TIME)
C        VALU   -  PARAMETER VALUE SPECIFYING POINT IN ORBIT            1545
C        ORBEL  -  ORBITAL ELEMENT VECTOR                               1546
C        STATE  -  STATE VECTOR AT SPECIFIED POINT                      1547
C                                                                       1548
C        WRITTEN 12/7/68                                                1549
C                                                                       1550
C     ***COMMONS***
      COMMON /CONCOM/
     1   PI,       SMU,      SLV,      SMP,      SKP,      RHODY,
     2   ACCS,     RCCS,     RHODY,    RHO2RHO,  TWOPI,    HALFPI
C     ***LOCALS***
                                                                  09/15/69
      DIMENSION ORBEL(10),STATE(10),AXES(10,3),SP(10),OE(10)          1552
C     EQUIVALENCE (SP(1),T),(SP(2),R),(SP(5),VR),(SP(6),VC)          66.0920
C     EQUIVALENCE (OE(1),P),(OE(6),E),(OE(7),TP),(OE(8),PP),(OE(9),A) 66.0920
      DATA (AXES(I,3),I=1,10)/10*0./                                  66.0920
      DATA (EMISSE=1.E-7)                                             66.0920

      [illegible]                                                     67.0811
      [illegible]                                                     67.0811
      [illegible]                                                     67.0811
      [illegible]                                                     67.0811
      [illegible]                                                     67.0811
      [illegible]
      CALL [illegible]                                                69.0709
      [illegible]

C     VALUE IS ANGLE
      [illegible]
      GO TO [illegible]
C     VALUE IS RADIUS
      [illegible]                                                     69.0321
```

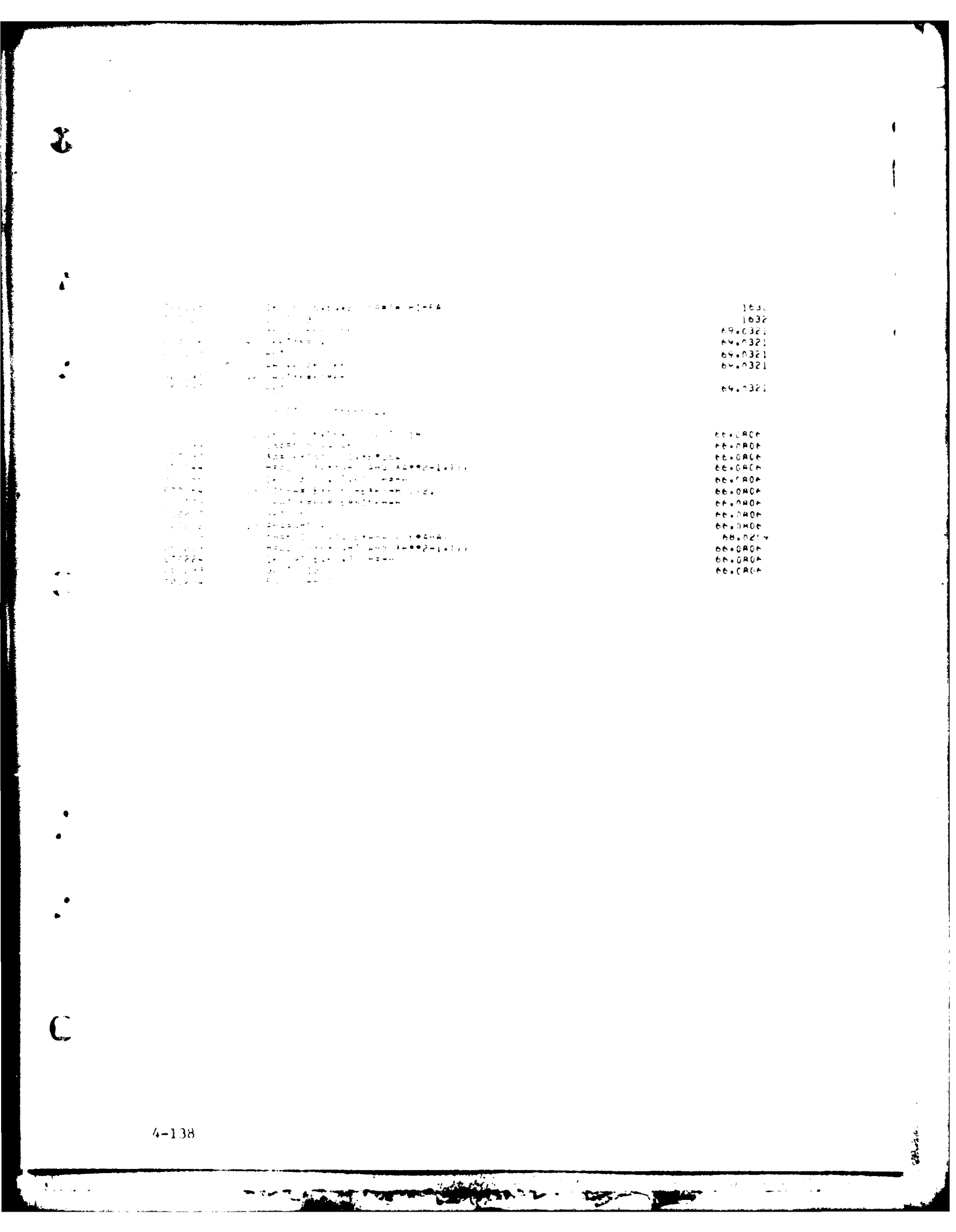
PRECEDING PAGE BLANK-NOT FILMED

```
                                                                  68.0014
[illegible]      [illegible] IF [illegible] VALUE,A,E,PR),TH
[illegible]      [illegible]
[illegible]      [illegible] (1.(R-R)/(R*E))                           68.0209
[illegible]      IF [illegible] E.F[illegible]) [illegible] = 0
C                        RADIUS IS DECREASING
[illegible]      [illegible]
                                                                  69.0321
C                [illegible]                                      69.0321
[illegible]   [illegible]                                         69.0321
000[illegible]      [illegible]                                   69.0321
[illegible]      [illegible]                                      69.0321
[illegible]      [illegible]                                      69.0321
C                [illegible]                                      69.0321
[illegible]   [illegible]                                         69.0321
000[illegible]      [illegible]                                   69.0321
[illegible]      [illegible]                                      69.0321 
[illegible]  [illegible] CALL [illegible] HAS REPLACED IMPOSSIBLE RADIUS WITH APOGEE/   69.0321
                 [illegible]                                      69.0321
[illegible]      [illegible]
C
C                VALUE IS TIME                                    66.0920
[illegible]   [illegible] =(VALUE-T[illegible])/[illegible]       66.0920
[illegible]      T=VALUE                                          66.0920
[illegible]      EA1=[illegible]                                  66.0920
[illegible]      [illegible]                                      66.0920
[illegible]   [illegible] =[illegible]+1                          66.0920
[illegible]      IF([illegible]) [illegible] 250                  66.0920
[illegible]      EA=[illegible]+E*SIN([illegible]EA1)             66.0920
[illegible]      IF(ABS(EA1-EA)/[illegible].EMISS) GO TO 42       66.0920
[illegible]      IF([illegible]) 45,46                            66.0920
[illegible]   45 EA1=EA2-(EA1-EA2)**2/(EA+EA2-2.*EA1)             66.0920
[illegible]      [illegible] 41                                   66.0920
[illegible]   46 EA2=EA1                                          66.0920
[illegible]      EA1=EA                                           66.0920
[illegible]      [illegible]                                      66.0920
[illegible]   42 [illegible]COS(EA)                               66.0920
[illegible]      [illegible]                                      67.1121
[illegible]      IF(A2.LE.[illegible])                            66.0920
[illegible]      A1=S[illegible](1.-E[illegible])*SIN(EA)         66.0920
[illegible]      [illegible]ATAN[illegible]A2)                    66.0920
[illegible]   [illegible] [illegible]                             66.0920
[illegible]      [illegible]

C                [illegible] ANSWER AND RETURN
[illegible]      [illegible]
[illegible]      [illegible]
[illegible]      [illegible]
[illegible]      CALL [illegible],AXES,[illegible]
[illegible]      CALL [illegible](STATE,AXES,SP,-1,1)

[illegible]      RETURN
C
C
C                ERROR MESSAGE
000[illegible]   [illegible] CALL [illegible]                                   69.0719
000[illegible]      CALL [illegible]
                                                                  69.0719
[illegible]      WRITE([illegible])                                           69.0719
[illegible]   [illegible] ELEMENTS ARE*/(5G20.8))                 66.0614
[illegible]      [illegible] HAS FAILED TO REACH A SOLUTION *)        67.0714
[illegible]      RETURN
[illegible]      END
```

```
      SUBROUTINE ORBPD(ORB,TIME,DSTAT)
C     SCHMIDT DATE AUG 714    TEST FOR APOGEE/PERIGEE
C     SCHMIDT DATE AUG 7[illegible]    LEAVE DSTAT ARRAY UN-NORMALIZED
*     SCHMIDT DATE AUG [illegible]        CONVERT TO CDC 6400
C     SCHMIDT DATE FEB [illegible]          BRAND NEW CODE
C
C     COMPUTES PARTIAL DERIVATIVES OF STATE VECTOR ELEMENTS (2-7)
C     WITH RESPECT TO ORBITAL ELEMENTS (2-7)
C
C     USE[illegible] [illegible] HYPERBOLIC ORBITS
C
C     ORB    -    ORBITAL ELEMENT ARRAY
C     TIME   -    TIME AT WHICH POINT IS REACHED
C     DSTAT  -    PARTIALS OF STATE AT GIVEN TIME
C
      [illegible]
      [illegible]
     1   [illegible]      SHE,       SLV,       SMF,       SKP,       RBODY,
     2   [illegible]      [illegible]       RBODY,     RBDZRO,    TWOPI,     RAFPI
      [illegible]
C                                                                   09/15/69
      DIMENSION ORBEL(10),DSTAT(6,6),S(10),ORB(10)
      EQUIVALENCE (ORBEL(2),EN1), (ORBEL(3),EN2), (ORBEL(4),EN3),
     1(ORBEL(5),P), (ORBEL(6),E), (ORBEL(7),TP), (ORBEL(8),PP),
     2(ORBEL(9),A)
C
      CALL XMIT([illegible],ORB,ORBEL)
      CALL CRBP(3,TIME,ORBEL,S)
      R=SQRT(S(2)**2+S(3)**2+S(4)**2)
      X=(P-R)/(R*E)                                                 69.0714
      IF(1.-[illegible]) [illegible]                                69.0714
    5 Q=ACOS(X)                                                     69.0714
      GO TO 12                                                      69.0714
    6 Q=PI                                                          69.0714
      GO TO 10                                                      69.0714
    [illegible] Q=0.                                                69.0714
      CALL [illegible] HAS REPLACED IMPOSSIBLE RADIUS WITH APOG     69.0714
     [illegible]EE/PERIGEE [illegible]                              69.0714
   [illegible] IF(E.[illegible]) GO TO 15                           69.0714
   [illegible] IF ((TIME-TP) [illegible]) Q=TWOPI-Q
      [illegible] 13
   15 IF ((TIME-TP) .LT. 0.) Q=TWOPI-Q
   [illegible] [illegible]SQRT(GC[illegible])
      [illegible]
      CQ=COS(Q)
      SQ=SIN(Q)
      C1=COS(EN1)
      S1=SIN(EN1)
      C2=COS(EN2)
      S2=SIN(EN2)
      C3=COS(EN3)
      S3=SIN(EN3)
      CT=COS(Q+EN3)
      ST=SIN(Q+EN3)
      [illegible]=-1.5*[illegible]*(TIME-TP)/A**2
      [illegible]E=((2.+E*CQ)*SQ-3.*E*RUP*(TIME-TP)/R**2)/(1.-E**2)
      [illegible]TP=-R[illegible]/R**2
```



```
      SUBROUTINE KEPTIM(E,VALUE,A,E,PP)                                  1599
C     SOURCE DATE AM.0361          TEST FOR APOGEE/PERIGEE
C     SOURCE DATE AM.1209        CONVERT TO CDC 6400
C     SOURCE DATE AM.1806   ALLOW HYPERBOLIC ORBITS
C     SOURCE DATE 66.1207
C                                                                        1600
C         RETURNS TIME SINCE PERIGEE PASSAGE AT POINT ON KEPLER ORBIT    1601
C
C         ELLIPTIC OR HYPERBOLIC ORBITS
C                                                                        1602
C         MODE   -  SELECTS SPECIFICATION OF POINT IN ORBIT              1603
C                   0  -  VALUE IS TRUE ANOMALY                          1604
C                   1  -  VALUE IS RADIUS (RADIUS INCREASING IN TIME)    1605
C                   2  -  VALUE IS RADIUS (RADIUS DECREASING IN TIME)    1606
C         VALUE  -  PARAMETER VALUE SPECIFYING POINT IN ORBIT            1607
C         A      -  SEMI-MAJOR AXIS OF ORBIT                             1608
C         S      -  NEGATIVE FOR HYPERBOLIC ORBITS
C         E      -  ECCENTRICITY OF ORBIT                                1609
C         P      -  GREATER THAN ONE FOR HYPERBOLIC ORBITS
C         PP     -  SEMI-LATUS RECTUM OF ORBIT
C         PE     -  ORBIT PERIOD DIVIDED BY 2 PI                         1610
C         TP     -  TIME OF PERIGEE PASSAGE
C                                                                        1611
C         WRITTEN 12/7/64                                                1612
C                                                                        1613
C     COMMON/CONST/
000030      COMMON /C KCON/
     1    PI,       SMU,      SLV,      SMP,      SKP,      RHODY,
     2    UASC,     XCON,     RHODY,    RHOZRO,   T=OPI,    HAFPI
C     COMMON/CONST/
                                                                  09/15/69
000110      DATA SLV S*1=0.9999999999)
C
000115      IF (MODE.LT.) GO TO 11                                         1615
C
C     ORBIT IS ELLIPTIC
C
000121      IF (E.NE.0.) GO TO 8                                           1616
000123      EA=VALUE
000124      CSA=COS(VALUE)                                                 1617
000132      IF(ABS(CSA).GT.ALMOST1) GO TO 3
000136      RADIUS=A*(1.-E*CSA)                                            1618
000144      IF (MODE.EQ.1.) REV=REV+1.                                     1619
000150      IF (VALUE.GT.0.) GO TO 5                                       1620
000154      EA=A*(SQRT(1.-E**2)*SIN(EA))/(1.-E*CSA)                        1621
000165      IF (VALUE.GT.T=OPI*REV.AND.EA.LT.T=OPI) EA=T=OPI-EA            1622
000204      EA=EA+T=OPI*REV                                                1623
000217      TIME=PP*(EA-E*SIN(EA))                                         1624
000227      RETURN                                                         1625
000230      SNA=SIN(VALUE)                                                 1626
000236      EA=VALUE+E*SNA*(1.-E**2*CSA)                                   1627
000247      GO TO 3                                                        1628
C                                                                        69.0321
C     VALUE IS RADIUS                                                    69.0321
000303      X=(VALUE-A)/(A*E)                                            69.0321
000307      IF(ABS(X).GT.1.) GO TO 12                                    69.0321
000312      EA=ACOS(-X)                                                  69.0321
```



```
C
C     FUNCTION ORBTIME RETURNS THE
C     SUBROUTINE ORBTIME(IEV,          ARRAY WHERE CODE AND VALUE
C
C     IEV - EVENT TYPE (1 - PERIGEE TIME AND VALUE
C
C     KODE - SPECIFIES INTERPRETATION OF VALUE
C            0 - VALUE IS TRUE ANOMALY
C            1 - VALUE IS MEAN ANOMALY (RADIANS INCREASING WITH TIME)
C            2 - VALUE IS MEAN ANOMALY (RADIANS DECREASING WITH TIME)
C            3 - VALUE IS TIME FROM PERIGEE TIME (VALUE)
C            4 - VALUE IS A TIME OF VALUE TIME INCREASING WITH TIME)
C            5 - VALUE IS A TIME OF VALUE TIME DECREASING WITH TIME)
C     VALUE - PARAMETER VALUE SPECIFYING POINT IN ORBIT
C     ORBIT - ORBITAL ELEMENT VECTOR
C
      SUBROUTINE ORBTIME(IEV,ARRAY,KODE,VALUE,ORBIT)
      COMMON /CONST/
     1   PI,        SQRT2,     SQ3,       SME,       SKE,       RADDEG,
     2   DEGRAD,    SQ2,       PHODEG,    PHOMIN,    TWOPI,     HALFPI
C
      DIMENSION ORBIT(12)
C
      ENTRY
      DIMENSION ARRAY(2)
      IF(M-3) 2,3,1
    1 WRITE(
      STOP
    2 ORBTIME = ORBTIM(M,I,ORBIT(9),ORBIT(10),ORBIT(11),ORBIT(12))
    3 RETURN
      END ORBTIME
```

```
      SUBROUTINE ORIO (TITLE,KFORM,NAMES,ORBEL,LINES)                    1636
C     SOURCE DATE AR.0206        CONVERT TO CDC 6400
C     SOURCE DATE AT.0726   CALL OLDATA ON READING SPECIAL DATA=CARD
C     SOURCE DATE AR.1222   ALLOW PUNCH OPTION
C     SOURCE DATE AR.0401    CALL OUTSET WITH 300 IN FIRST ARG
C                                                                        1637
C     READS AND PRINTS ORBITAL ELEMENTS                                  1638
C                                                                        1639
C     TITLE  =  TITLE FOR OUTPUT                                         1640
C     KFORM  =  OUTPUT FORMAT CONTROL                                    1641
C     NAMES  =  ARRAY OF FIVE-WORD NAMES FOR VECTORS IN ORBEL            1642
C     ORBEL  =  ARRAY OF 10-WORD ORBITAL ELEMENT VECTORS                 1643
C     LINES  =  NUMBER OF VECTORS IN ORBEL                               1644
C                                                                        1645
C     KFORM CONSISTS OF THREE DIGITS KR, KN, AND KI OPERATING THUS -     1646
C          KR=0  =  READING OMITTED                                      1647
C             1  =  READING INCLUDED                                     1648
C             2  =  PUNCHING ONLY                                66.1222
C          KN=0  =  NAMES IGNORED                                        1649
C             1  =  NAMES INCLUDED IN INPUT AND OUTPUT                   1650
C             2  =  AS KN=1, EXCEPT INPUT NAMES NOT RETURNED IN NAMES    1651
C          KI=0  =  IDENTIFICATION WORD IGNORED                          1652
C             1  =  IDENTIFICATION TAKEN AS TIME                         1653
C             2  =  IDENTIFICATION TAKEN AS HOLLERITH WORD               1654
C             3  =  NUMBER OF VECTOR IN ORBEL USED AS IDENTIFICATION     1655
C                                                                        1656
C     IF LINES IS NEGATIVE, TITLE AND COLUMN HEADINGS WILL BE            1657
C     OMITTED IN OUTPUT                                                  1658
C                                                                        1659
C     EACH INPUT ORBIT IS ENTERED ON TWO CARDS AS FOLLOWS -              1660
C                                                                        1661
C          COLUMN     FORMAT     CONTENT                                1662
C          ------     ------     -------------------------------------  1663
C     CARD 1                                                             1664
C          1-30       3A8,A6     HOLLERITH NAME
C          31-40      F10.0      IDENTIFICATION (FORMAT A8,2X IF KI=2)
C          41-50      F10.0      LONGITUDE OF ASCENDING NODE, DEG.      1667
C          51-60      F10.0      INCLINATION, DEG.                      1668
C          61-70      F10.0      ARGUMENT OF PERIGEE, DEG.              1669
C     CARD 2                                                             1670
C          1-30       3A8        EPOCH                                  1671
C          31-40      F10.0      SEMI-LATUS RECTUM, KM OR N MI **       1672
C          41-50      F10.0      ECCENTRICITY                           1673
C          51-60      F10.0      TIME OF PERIGEE PASSAGE, SECONDS       1674
C          61-70      F10.0      SEMI-MAJOR AXIS, KM OR N MI **         1675
C                                                                        1676
C          **  =  IF EITHER IS BLANK OR ZERO, THE OTHER SERVES TO       1677
C                 ESTABLISH THE ORBIT, OTHERWISE THE FORMER SERVES.      1678
C                                                                        1679
C     WRITTEN 12/6/64                                                    1680
C                                                                        1681
      COMPASS,BASCON
000006      COMMON /BASCON/
           1   PROGRM,   KPAGE,    LINE,     IODAY,    RUN,      PUNTO(6),
           2   MSG,      FLAG,     DATE,     MAXTM,    MAXOG,    MAXLN,
           3   KLASS,    KGROUP,   KUNITS,   KSTINT,   KOCHD,    IEOF
           4  ,NOFILE
C
      COMPASS,CONCON
000006      COMMON /CONCON/
           1   PI,       SRD,      SLV,      SME,      SKP,      RBODY,
           2   G&CC,     GCON,     WBODY,    RHO7RO,   TWOPI,    HAFPI
C
      COMPASS,STACON
000006      COMMON /STACON/
           1   KFMT,     KODES(6), MODES(4), CPTNS(10,8),          SETU@(82)
C
```

PRECEDING PAGE BLANK-NOT FILMED

```
000016          DIMENSION NAMES(5,2),JWREL(10,2),QI(10),NAME(5),H(82),UNITS(2,4,2)   66,1222
000006          EQUIVALENCE (UNITS,CPTNS(85))                                        1687
000006          EQUIVALENCE (AEQ,IEQ)
000006          DATA IKEY=0/
000006          NAMES=IAHS(INES)                                                     1725
000010          KUNITS=0                                                             1688
000012          KEY=IAHS(KEYWD)                                                      1689
000013          KQ=MOD(KEY/100,10)                                                   1690
000021          KWD=MOD(KEY/10,10)                                                   1691
000026          IF(KQ.GT.0.AND.KQ.LT.3)                                           66,1222
000031          KI=MOD(KEY,10)                                                       1692
000034          IF(KEY.EQ.KERMIT.OR.KEY.EQ.11)
000036          KERMIT=KEY
000038          I=12*
000040          IF (KWD.NE.0) GO TO 6                                             66,0601
000041          WRITE(6,7)                                                           1697
000043          FORMAT(1H0,6X,7H *
000052    5     WRITE(6,8)                                                           1699
000053          FORMAT(1H0,6X,7HSET AT,2H *,I2H    TIME    ,2H *,I2H             ,2H *,I2H      SEC     ,2H*) 67,0726
000060          GO TO 9                                                              1701
000064    6     WRITE(6,8)                                                           1702
000065          WRITE(6,7)                                                           1703
000066    7     FORMAT(1H0,6X,
000067    8     FORMAT(1H0,6X,7HSET AT,2H *,I2H                ,2H *,I2H             ,2H*)    1705
000075    9     WRITE(6,10)
000101   10     FORMAT(1H *
000102          FORMAT(1H0,6X,7HSET AT,2H *,I2H  ANGULAR RATE  ,2H *,I2HASCENDING NODE,2H *,UNITS(1,4,KU),2H*)  67,0726
000140          WRITE(6,7)                                                           1710
000136          FORMAT(1H0,6X,7HSET AT,2H *,I2H  APOGEE        ,2H *,I2H  PERIGEE    ,2H *,UNITS(1,4,KU),2H*)  67,0724
000151          WRITE(6,7)                                                           1712
000162          FORMAT(1H0,6X,7HSET AT,2H *,I2H  SEMI-MAJOR AXIS ,2H *,I2HLATUS RECTUM,2H *,UNITS(1,1,KU),2H*)   1713
000174          WRITE(6,7)                                                           1714
000205          FORMAT(1H0,6X,7HSET AT,2H *,I2H  ECCENTRICITY  ,2H *,I2H  TRICITY     ,2H *,I2H              ,2H*)  1715
000213          WRITE(6,100)                                                         1716
000226          FORMAT(1H0,6X,7HSET AT,2H *,I2H  TIME AT       ,2H *,I2H  PERIGEE     ,2H *,I2H      SEC     ,2H*) 67,0724
000235          IF(KEY.EQ.0) GO TO 11                                                1718
000241          WRITE(6,7)                                                           1719
000242          FORMAT(1H0,6X,7HSET AT,2H *,I2H  DE TOF        ,2H *,I2H              ,2H *,I2H      SEC     ,2H*) 67,0724
000271          IF(KEY.EQ.1) GO TO 11                                                1721
000272          WRITE(6,7)                                                           1722
000304          FORMAT(1H0,6X,7HSET AT,2H *,I2H  SEMI-MAJOR AXIS ,2H *,I2H WINDOW AXIS ,UNITS(1,1,KU),2H*)   1723
000324    11    IF (INES.GE.1.AND.INES.LE.5) TOPLOT(6H*NAMES,I,0)                    1724
000260          GO TO (12,13,14,15,16),INES                                          1724
000262          IF (KWD.EQ.0) GO TO 17                                               1727
000263          IF (KWD.EQ.2) GO TO 18                                               1728
000265          READ (5,2) (NAME(I,I=1,4),I=1,8)                                     1729
000334    12    FORMAT(32A4,4A1,4F10.0/31A1,F10.0)                                   1730
000340          GO TO 15                                                             1731
000310    13    READ (5,14) (NAME(I,I=1,4),I=1,8)                                    1732
000330    14    FORMAT(32A4,4A1,2X,3F10.0/30X,F10.0)                              67,0726
000333    15    IF(KEY.EQ.0.OR.KEY.EQ.1) GET_DV                                   68,0209
000341          IF (NAME(1,1).NE.IWHO) DATA_2=DATA                                67,0726
000361          DATA=TODATA(4HDRIC)                                               67,0726
000364          GET_DV                                                            67,0724
000364    16    DO 17 I=1,4
000351    17    NAME(I)=SUN(I)                                                       1735
000356          NAME(5)=SOL                                                          1736
000357          NAME(6)=MOON                                                         1737
000361          IF (KWD.EQ.0.AND.KEY.EQ.5) AHS(2/9)=1.-ECC(A)**2)                    1738
000365          IF (KWD.EQ.0.AND.KEY.EQ.5) AHS(2/3)=1.-ECC(A)**2))                   1739
000372          ANG(9)=SQRT(1/9)/GCON)                                               1740
000400          I=1
000401          DO 18 I=1,10                                                         1742
000405    18    WRITE(7,*) YNE=NAME(I)                                               1743
000426          IF (KWD.NE.2) GO TO 20                                               1744
000420          DO 19 I=1,5                                                          1745
000422    19    NAMES(I,1)=NAME(I)                                                   1746
000432    20    DO 21 I=1,10                                                         1747
000434    21    WRITE(6,22)(NAME(I,J),J=1,4),YNE)                                    1748
000445          DO 22 I=2,4                                                          1749
000447    22    WRITE(6,23)(NAME(I,J),J=1,4),YNE)                                    1750
000450          ANG(5)=ANG(5)/SLV                                                    1751
000455          WRITE(7,*)WOP(9)
000457          ANG(9)=ANG(9)/SLV                                                    1753
000460          IF (KWD.EQ.2) GO TO 24                                               1754
000462          DO 23 I=1,5                                                          1755
000463    23    WRITE(7,*)NAMES(I,1),YNE)                                            1756
000473    24    FORMAT
```

```
000475          IF(KI.EQ.3) CALL EXIT
000500          KK=1
000501          IF(KI.EQ.0) KK=2
000503          CALL OUTPUT(TITLE,M,NAME,ORIG,KK)
000511    25    CONTINUE                                                    1759
000517          RETURN                                                      1760
                C
000517       30 CALL XMIT(-5,1H ,NAME)                                      A6.1222
000522          DO 32 L=1,MAES                                              A6.1222
000527          IF(KK.GT.1) CALL XMIT(5,NAME(L),NAME)                       A6.1222
000544          CALL XMIT(-5,3HAES(I,L),3)                                  A6.1222
000567           A2(L)=A2(L)*SR2                                            A6.1222
000611           A3(L)=A3(L)*SR3                                            A6.1222
000632           A4(L)=A4(L)*SR4                                            A6.1222
000653           A5(L)=A5(L)/SL                                             A6.1222
000674           A6(L)=A6(L)/SL                                             A6.1222
000716       32 WRITE(A2,33) NAME(I),(A2(L),A3(L),A4(L),A5(L),A6(L),L=1,M)
000803       33 FORMAT(3AH,A4,F12.3,3F10.5/30(F12.5,5F10.8,F10.2,F10.4))    A6.1222
000803          RETURN                                                      A6.1222
000803          END
```

```
      SUBROUTINE OUT1(TEXT,VALUE,MPRINT)
C     SOURCE DATE 68.0819    RESET K AT END OF LOOP
C     SOURCE DATE 68.0209    CONVERT TO CDC 6600
C     SOURCE DATE 66.0101    HRAND NEW CODE
C
C     PRINTS A SHORT DESCRIPTION AND ONE PIECE OF DATA
C
C          TEXT -    HOL DESCRIPTION OF DATA BEING PRINTED,
C                    UP TO 40 CHARACTERS LONG, TERMINATED BY A DASH
C          VALUE -   THE VARIABLE TO BE PRINTED
C          MPRINT -  CODE DEFINING PRINT-FORMAT,
C                    AS DESCRIBED IN SUBR OUTSET
C
C                                  1    2    3    4    5    6    7    8    9    0
C          MPRINT          =
C          CONVERSION      =  E.4  F.2  F.6  F.3  F.0  I16  I16  O16  A14  A16
C
C
      COMPXC,HASCON
      COMMON /RASCON/
     1    PROGRM,     KPAGE,     LINE,      TOFDAY,    RUN,       QUINTD(6),
     2    MSG,        FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
     3    KLASS,      KGROUP,    KUNITS,    KSTINT,    KOORD,     IFEOF
     4    ,NOFILE
000005      DIMENSION FORM(10),TEXT(1),IMAGE(2),TEX(4)                        68.0209
000005      DATA (FORM=5HM(E16.4    (E16.2)    (F16.6)    (F16.3    (F16.0)  ,68.0209
     *              5HM(I16)      (I16)      (O16)      (A16)     (A16)    )68.0209
000005      CALL XMIT(6,TEXT,TEX)                                             68.0209
000005      K=0                                                               68.0209
000010      DO 12 K=1,39                                                      68.0209
000011      CALL MCHAR(K,TEXT,10,M1)                                          68.0209
000015      IF(M1.EQ.10) GO TO 14                                             68.0209
000021   12 CONTINUE                                                          68.0209
000025      K=31                                                              68.0209
000027      14 NN=(K+9)/10                                                    68.0819
000030      IF(NN.LT.4) CALL XMIT(NN-4,1H ,TEX(NN+1))                         68.0209
000034
000045      ENCODE(16,FORM(MPRINT),IMAGE)VAL F                                68.0209
000050      LINE = LINE+1
000060      IF(LINE.GT.MAXLN) CALL HEAD(2)
000066      WRITE(6,1) TEX,IMAGE
000076    1 FORMAT(15X,A10,4H=  ,A10,4H)
000076      RETURN
000077      END  OUT1
```

PRECEDING PAGE BLANK-NOT FILMED

```
                                                                        1762
                                                                        1763
                                                                        1764
                                                                        1765
                                                                        1766
                                                                        1767
                                                                        1768
                                                                        1769
                                                                        1770
                                                                        1771
                                                                        1772
                                                                        1773
                                                                        1774
                                                                        1775
                                                                        1776
                                                                        1777
                                                                    09/15/69
                                                                        69.0709
                                                                        69.0709
                                                                        69.0709
                                                                        69.0709
                                                                    09/15/69
                                                                        1780
                                                                        1783
                                                                        1784
                                                                        1785
                                                                        1786
                                                                        1787
                                                                        1788
                                                                        1789
                                                                        1791
                                                                        1792
                                                                        1794
                                                                        1795
```

```
                                                                        1796
      KOLUMS=0
      DO 7 I=2,20,2
      IF (SETUP(I),EQ.PFORMS(1,10)) KOLUMS=KOLUMS+1
      IF (SETUP(I),EQ.SPACE) KOLUMS=KOLUMS+1                            1800
    7 CONTINUE
      NAMEC=0                                                           1802
      DO 9 I=2,4
      IF (SETUP(2*I),EQ.PFORMS(1,I+9)) GO TO 9
      NAME=I+1
      KOLUMS=KOLUMS+1
    9 CONTINUE
      IF (LYNES.EQ.0) RETURN                                            1806
      DO 12 LYNE=1,LYNES                                                1807
      IF (LINE.LT.SKALEN) GO TO 12                                      1808
      CALL PAGE(3)                                                      1809
      IF (SETUP(6).EQ.0) GO TO 12                                       1810
      WRITE (6,11) (SETUP(I),I=23,LAST)                                 1811
   11 FORMAT(1H+,T(A10,A2)/10(A10,A2))                                  6H.0204
      LINE=LINE+SETUP(6)                                                1813
      CALL LSPACE                                                       1814
      LINE=LINE+1                                                       1815
      I=KOLUMS                                                          1816
      II=I*KOLUMS+1                                                     1817
      IF (NAME.EQ.1) GO TO 18                                           1818
      WRITE (6,15) (NAMES(I,LYNE),I=1,NAME),(DATA(I),I=II,12)
      GO TO 21                                                          1820
   18 WRITE (6,19) (DATA(I),I=II,12)                                    1821
   21 CONTINUE                                                          1822
      RETURN                                                            1823
      END
```

```
      SUBR  TO E OUTN  TEXT,VALUE,MPRINT,NVAL)
C     SOURCE DATE AR,0419   RESET K AT END OF LOOP, BLANK OUT IMAGE
C     SOURCE DATE AR,0209       CONVERT TO CDC AA60
C     SOURCE DATE AA,0101   BRANCH NEW CODE
C
C     PRINTS A SHORT DESCRIPTION AND SEVERAL PIECES OF DATA
C
C          TEXT =    BCD DESCRIPTION OF DATA BEING PRINTED,
C                    UP TO 40 CHARACTERS LONG, TERMINATED BY A DASH
C          VALUE =   FIRST LOCATION OF THE STRING OF VARIABLES
C                    TO BE PRINTED
C          MPRINT =  CODE DEFINING PRINT-FORMAT,
C                    AS DESCRIBED IN SUBR OUTSET
C          NVAL =    NUMBER OF VARIABLES IN THE STRING,
C                    IF NVAL POSITIVE, VALUES ARE PRINTED LEFT TO RIGHT,
C                                 FIVE VALUES TO A LINE
C                    IF NVAL NEGATIVE, VALUES ARE ARRANGED IN 5 COLUMNS
C
C          MPRINT     *   1    2    3    4    5    6    7    8    9    0
C          CONVERSION *  E**  F.2  F.6  F.3  F.0  I1A  I16  O16  A14  A16
C
      COMMON/RASCON/
000000      COMMON /RASCON/
     1   PLOTPM,   KPAGE,    LINE,     TOFDAY,   RUN,      RUNID(6),
     2   MSG,      FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
     3   KLASS,    KGROUP,   KUNITS,   KSTINT,   KOORD,    IFEOF
     4   /NOFILE
C
000006      DIMENSION IMAGE(2, 5),VALUE(NVAL)
                                                                        68.0209
000006      DIMENSION FORM(10),TEXT(1),TEX(6)                           68.0209
000034      DATA (FORM=5HM=(E16.4    F16.2    (F16.6    (F16.3    (F16.0    ,  68.0209
     *            5HM(I16)     (I16)     (O16)     (A16)     (A16)     )   68.0209
000006      DATA (ABO=2HBO),(BLK=1H )                                       68.0819
C
000006    1 FORMAT(1HXAA10,2(  *  5(A10,2A) ))                              68.0209
000016    2 FORMAT(53X R1, 2X     5(A10,2A) )                               68.0209
C
000026      CALL XMIT(6,TEXT,TEX)                                           68.0209
000031      K=1                                                             68.0209
000034        IF (K=1,30                                                    68.0209
000040      CALL MOVARY(K,TEXT,10,M,1)                                      68.0209
000042      IF(M.EQ.1)R=1 GO TO 14                                          68.0209
000047    1 CONTINUE                                                        68.0209
000051      M=1                                                             68.0819
000052    14 NN=K+9)/10                                                     68.0209
000036      IF(NVAL.LT.0) CALL XMIT(NN+4,IN ,TEX(NK+1))                     68.0209
C
000055      GREATER=VAL
000063      LINE=0/5+1/5
000057      IF(LINE+NLIN.GT.MAXLN) CALL HEAD(2)                             68.0819
000065      LINE=LINE+NLIN
000070      IF(NVAL.GT.0) GO TO 30
C
000072        J 22 N=1,NLIN
000073      CALL XMIT(-10,BLK,IMAGE)                                        68.0819
000075      K=0
000076      DO 18 MM=1    ,NV,NLIN
000103      K=K+1
000105    18 ENCODE(10,FORM(MPRINT),IMAGE(1,K)) VALUE(MM)
000132      IF(N.GT.1) GO TO 20
000136      WRITE(6,1) TEX,IMAGE
000145      GO TO 22
000150    23 WRITE(6,2) ABO,IMAGE                                           68.0209
000160    22 CONTINUE
000165      RETURN
C
000166    30 DO 36 N=1,NLIN
000170      CALL XMIT(-10,BLK,IMAGE)                                        68.0819
```

```
000172          MA=5*(N-1)+1
000175          MR=MIN0(MA+4,NV)
000201          DO 32 MM=MA,MR
000206          MM=MM-MA+1
000210       32 ENCODE(14,FORM(MPRINT),IMAGE(1,M)) VALUE(MM)
000235          IF(N.GT.1) GO TO 34
000241          WRITE(6,1) TFS,IMAGE
000250          GO TO 36
000253       34 WRITE(6,2) BLK,IMAGE                                          68,0209
000263       36 CONTINUE
000270          RETURN
000271          END (0,1)N
```

```
      S[illegible] SETUP(LINE1,LINE2,LINE3,SETUP)                          1825
C     [illegible]  WIDEN F.3 AND F.6 FORMAT-FIELDS
C     [illegible]  MAY HAVE 20-, 30-, OR 40-CHARACTER NAMES
C                                                                          1826
C     SETS UP A COLUMN HEADING AND PRINT FORMAT FOR USE IN OUTCOL          1827
C                                                                          1828
C     MODE   =  A CODE WORD OF 3 DIGITS MSTART, MCOL, MPRINT               1829
C               MSTART = CONTROL FOR INITIALIZATION OF SETUP               1830
C                    0 = NO INITIALIZATION                                 1831
C                    1 = INITIALIZATION FOR 10 BLANK COLUMNS               1832
C                    2 = INITIALIZATION FOR 10 BLANK COLUMNS WITH
C                        20-CHARACTER LINE NAME IN COLS. 1-2.              66.0601
C                    3 = INITIALIZATION FOR 10 BLANK COLUMNS WITH          66.0601
C                        30-CHARACTER LINE NAME IN COLS 1-3.               66.0601
C                    4 = INITIALIZATION FOR 10 BLANK COLUMNS               66.0601
C                        40-CHARACTER LINE NAME IN COLS. 1-4.              66.0601
C               MCOL = NUMBER OF COLUMN NOW TO BE SET UP                   1835
C               MPRINT = PRINT FORMAT FOR DATA IN COLUMN                   1836
C                    1  =  E12.4
C                    2  =  E11.2
C                    3  =  F11.6                                           69.0709
C                    4  =  F11.3                                           69.0709
C                    5  =  F10.0
C                    6  =  I8
C                    7  =  I10
C                    8  =  O12                                             1844
C                    9  =  A10
C                    0  =  A8,A2
C     LINE1  =  12-CHARACTER, FIRST-LINE COLUMN HEADING                    1847
C     LINE2  =  12-CHARACTER, SECOND-LINE COLUMN HEADING                   1848
C     LINE3  =  12-CHARACTER, THIRD-LINE COLUMN HEADING                    1849
C     SETUP  =  ARRAY OF COLUMN FORMATS AND HEADINGS FOR USE IN            1850
C               OUTCOL, SET UP BY REPEATED CALLING OF OUTSET               1851
C                                                                          1852
C     NOTE 1 =  SINCE COLUMNS ARE 12 CHARACTERS WIDE WITH NO SPACES        1853
C               IN BETWEEN, SPACE BETWEEN ADJACENT COLUMN HEADINGS         1854
C               MUST BE PROVIDED (AS DESIRED) IN THEIR SPECS               1855
C     NOTE 2 =  SINCE LEFTMOST CHARACTER OF COLUMN 1 HEADS APPEARS         1856
C               IN PRINTER CARRIAGE CONTROL COLUMN, IT MUST BE BLANK       1857
C                                                                          1858
C     WRITTEN 6/5/64                                                       1859
C                                                                          1860
C     [illegible],FORMATS                                                  69.0709
      COMMON /FORMATS/ FORMS(2,13)
      DATA (FORMS =
     1 12H-12.5,      ,12H-G11.4,1X     ,12HF11.6,1X      ,              69.0709
     2 12HF11.3,1X    ,12HF10.0,2X      ,12HI8,4X         ,              69.0709
     3 12HI10,2X      ,12HO12,          ,12HXA10,X        ,              68.0209
     4 12H(1X,8H,A2,1X,12H2X2A8,A4,2X   ,12H3X3A8,A6,3X  ,
     5 12H(6A8,4X     )
C     [illegible],FORMATS                                                  69.0709
C                                                                        09/15/69
      DIMENSION SETUP(82),LINE1(2),LINE2(2),LINE3(2)
      DATA (PAREN=1H(),(PARENR=1H)),(SPACE=3H12X),(BLANK=1H )
      REAL LINE1,LINE2,LINE3

      MSTART=MOD MODE/100,10)
```

```
                                                                        1868
                                                                        1869
                                                                        1870
                                                                        1871
                                                                        1873
                                                                        1874
                                                                        1875
                                                                        1876
                                                                        1877
                                                                        1878
                                                                        1879
                                                                        1886
                                                                        1887
                                                                        1888
                                                                        1889
                                                                        1893
```

```
                                IDENT   PACBIT
                      000037    PROGRAM LENGTH

                                BLOCKS

                      000037    PROGRAM#  LOCAL

                                ENTRY POINTS

                      000000 PACA          000001 UNPA          000002 BITA

                                ENTRY   PACA
                                ENTRY   UNPA
                                ENTRY   BITA
                                *
                                *       FOR ENTRY PACA, TAKES APPROPRIATE BITS FROM RIGHT END OF KERNEL
                                *       AND PACKS THEM INTO BITS II THRU JJ OF WORDS, LEAVING THE REST OF
                                *       WORDS UNDISTURBED.
                                *
                                *       FOR ENTRIES UNPA OR BITA, UNPACKS THE APPROPRIATE BITS FROM
                                *       BITS II THRU JJ OF WORDS, AND STORES IN KERNEL, FILLING THE REST
                                *       OF KERNEL WITH ZEROS.
                                *
                                *       CALL PACA OR (UNPA=OR=BITA)  WITH ARGUMENTS ...
                                *                       (KERNEL,II,JJ,WORDS)
                                *       JJ MUST NOT BE GREATER THAN II+59 (NOT MORE THAN ONE WORD WIDE
                                *
                                *       PERMANENT REGISTER ASSIGNMENTS
                                *       A1      ADDR(KERNEL)
                                *       A2      VALUE(II)
                                *       A3      VALUE(JJ)
                                *       A4      ADDR(WORD(K))
                                *       A5      A0
                                *       A7      FLAG DENOTING ENTRY TO FOR BITA, 1 FOR PACA)
                                *       A0      MASK
                                *       X1      LEFT-BIT-NUMBER OF MASK
                                *       X2      RIGHT-BIT-NUMBER OF MASK
                                *       X3      IF POSITIVE, KERNEL IS SPLIT OVER TWO WORDS
                                *
                      PACA      EQU     1
                      UNPA      EQU     0
                                SET     1
                                JP      .1
                                *
                      BITA      EQU     1
                                SB7     .
                      .1        SA2     B2
                                SB2     X2
                                SA3     A3
                                SB3     X3
                                SB6     A0
                      .2        LE      A2,B6,.3
                                SB2     A2-B6
                                SB3     A3-B6
                                SB4     A4+1
                                JP      .2
                                *
```

```
                SA1     B6
                SA2     B3
                SA3     X2-B1
                BX      X3*X4
                SA2     B0
*
                RJ      GETMASK
*
                SB5     B6-B3
                PL      B5,*5
                SB5     B5*B4
*
                ZR      B7,BIT1
PAC1            SA5     B1
                LX5     X5,B5
*
PAC2            SA4     B0
                BX4     -X0*X4
                BX6     X0*X5
                BX6     X6+X4
                SA6     A4
                BX      X3+PACX
                JP      *6
*
BIT1            SA4     B4
                BX6     X6*X4
                SB5     B6-B5
                LX6     X6,B5
                PL      X3,*6
                SA6     B1
                JP      BITX
*
*6              SB3     B3-B6
                SB4     B6+1
                SA1     1
                SA2     B3
                RJ      GETMASK
*
*7              ZR      B7,BIT3
*
PAC3            SA3     -1
                JP      PAC2
*
BIT3            SA4     B4
                BX7     X6*X4
                LX7     X7,B3
                BX6     X6+X7
                SA6     B1
                JP      BITX
*
*
*               GETMASK RA  INPUTS ... X1+X2+BA
*               GETMASK HAS OUTPUTS .. A0
*               GETMASK USES ......... X7+B5
GETMASK         MASZ    1
                MA0     1
                IX7     X6-X1
                SB5     X7
*
                AA0     X6+B5
                SB5     X1
                SB5     B6-B5
                SB5     B5+1
                LA0     X6,B5
                JP      GETMASK
*
                END

022536      UNUSED STORAGE          112 STATEMENTS      14 SYMBOLS
```

```
                      SUBROUTINE PLOTT(TITLE,X0,Y0,X1,Y1,DATA,KOLUMS,LINES,KX,KY)                1895
                      SOURCE DATE A3.0301
C                                                                                                1896
C       SKIPS TO NEW PAGE, HEADS IT, PLOTS UP TO SIX Y TRACES VS. X.                             1897
C       PLACES TITLE BELOW GRAPH UNLESS TITLE IS BLANK.                                          1898
C                                                                                                1899
C       ORIGIN OF PLOT IS AT (X0,Y0).  PLOT EXTENDS TO (X1,Y1).                                  1900
C       (PLOTTED REGION IS 100 PRINT COLUMNS BY 50 PRINT LINES)                                  1901
C                                                                                                1902
C       DATA CONTAINS LINES ROWS OF KOLUMS COLUMNS EACH.  PLOTT                                  1903
C       CAN PLOT ANY SIX COLUMNS VS ANY OTHER COLUMN ACCORDING TO                                1904
C       THE SELECTIONS INDICATED IN KX AND KY.  KX IS THE NUMBER OF                              1905
C       THE COLUMN TO BE PLOTTED ALONG THE HORIZONTAL AXIS X.  IF                                1906
C       PRECEDED BY A MINUS, VERTICAL BARS (HISTOGRAMS) ARE PRO-                                 1907
C       DUCED.  KY CONTAINS UP TO SIX DIGITS, EACH OF WHICH INDI-                                1908
C       CATE A COLUMN NUMBER TO BE PLOTTED IN THE VERTICAL DIRECTION.                            1909
C       IF KX IS ZERO, THE LINE NUMBER IS USED AS X-COORDINATE                                   1910
C                                                                                                1911
C       POINTS OUTSIDE THE PLOTTED REGION ARE IGNORED.  FOR A 57-LINE                            1912
C       PAGE, 3 LINES ARE AVAILABLE BELOW THE PLOT FOR A TITLE IF                                1913
C       ARGUMENT TITLE WAS BLANK.                                                                1914
C                                                                                                1915
C       SYMBOLS USED IN THE PLOT ARE AS FOLLOWS -                                                1916
C                                                                                                1917
C                 TRACE 1 -  *        TRACE 4 -  0                                           1918
C                 TRACE 2 -  +        TRACE 5 -  @                                           1919
C                 TRACE 3 -  X        TRACE 6 -  =                                           1920
C                                                                                                1921
C       WRITTEN 3/63                                                                             1922
C
      COMPASS,BASCOM
000013        COMMON /BASCOM/
             1   PROGRM,   KPAGE,    LINE,     TOFDAY,   RUN,      RUNID(6),
             2   MSG,      FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
             3   CLASS,    KGROUP,   KUNITS,   KSTINT,   KOORD,    IFEOF
             4   NFILE
C
000013        DIMENSION DATA(KOLUMS,LINES),PONT(100),XL(5),YL(5),SYMBOL(7)                       1925
000013        EQUIVALENCE (SYMBOL(7),BLANK)                                                      1926
000013        DATA SYMBOL(1),I=1,7)/1H*,1H+,1HX,1H0,1H@,1H=,1H /                                 1927
C
000013        XX=IABS(KX)                                                                        1928
000015        CALL HEADIT                                                                        1929
000017          XM=(X1-X0)/100.                                                                  1930
000026          YM=(Y1-Y0)/100.                                                                  1931
000036          DO 2 I=1,5                                                                       1932
000032          XL(I)=0.                                                                         1933
000035          YL(I)=0.                                                                         1934
000041    2     CONTINUE                                                                         1935
000044          DO 13 JJ=2,100,2                                                                 1936
000047          J=JJ-1                                                                           1937
000051          XJ=J                                                                             1938
000054          IF (KX.LT.0.AND.JJ.NE.2) GO TO 4                                                 1939
000064          DO 3 I=1,100                                                                     1940
000065    3     PONT(I)=BLANK                                                                    1941
000071    4     SYMBOL=0                                                                         1942
000072          DO 9 K=1,6                                                                       1943
000074          IF (KY.LT.10**(6-K)) GO TO 9                                                     1944
000103          KOLUM=MOD(KY/(10**(6-K)),10)                                                     1945
000115          IF (IABS(KY).GE.10) KXX=MOD(IABS(KX)/(10**(6-K)),10)                             1946
000134          SYMBOL=NSYMB(K)                                                                  1947
000136          IF (KOLUM.EQ.0) KOLUM=10                                                         1948
000140          DO 8 LYNE=1,LINES                                                                1949
000142          IF (ABS(DATA(KOLUM,LYNE)-YJ).GT.-Y) GO TO 8                                      1950
000153          IF (ABS(DATA(KOLUM,LYNE)-YJ).LT.0Y) GO TO 8                                      1951
000163          IF (DATA(KOLUM,LYNE)-YJ.GE.0.) GO TO 8                                           1952
000171    5     IX=(DATA(KXX,LYNE)-X0)/XM+.5                                                     1953
000205          IF (KX.EQ.0) IX=LYNE                                                             1954
000210          IF (IX.LE.0) GO TO 8                                                             1955
```

```
000212      6     IF (IX.GT.100) GO TO 8                                                    1956
000216      7     POINT(I)=SYMBOL(NSYMBL)                                                   1957
000222      8     CONTINUE                                                                  1958
000225      9     CONTINUE                                                                  1959
000227            IF (MOD(I,20).NE.0) GO TO 11                                              1960
000233            WRITE                                                                     1961
000235            WRITE (6,10) YL(I),(POINT(L),L=1,100)                                     1962
000261     10     FORMAT(3X   F10.3, 2X1H* 100A1)                                           1963
000261            GO TO 13                                                                  1964
000265     11     WRITE (6,12) (POINT(L),L=1,100)                                           1965
000303     12     FORMAT(15X 1HI 100A1)                                                     1966
000303     13     CONTINUE                                                                  1967
000311            WRITE (6,14)  YL(I),(I=1,5)                                               1968
000326     14     FORMAT(15X 1H+ 1(10H----------) /20X   5(10XF10.3))                       1969
000326     15      INE=LINE+5                                                               1970
000330            CALL TITLER(15,TITLE)                                                     1971
000336     99     RETURN                                                                    1972
000337            END  PLOTT
```

```
$ENTRY PREDATA
* AUTHOR: R.E. ANDERSON   LARGELY REWRITTEN, TO ADD UPTO,INCL,BY,EXEC
* DATE:  OCT 28,1965   BRAND NEW CODE

* PREDATA PRINTS THE IMAGES OF DATA-CARDS.
*  PREDATA MUST BE CALLED BEFORE THE INITIAL CALL OF READ, AND
*  THE 10 BUFFERS MUST BE OF THE FORM ... (INPUT,TAPES,...)
*  $ENTRY(INPUT)
*  EACH CALL TRANSFERS DATA-CARDS FROM INPUT TO TAPES, UNTIL
*  IT FINDS AN EOF OR A MONDATA CARD.

* MONDATA EDITS THE DATA-CARDS ON TAPES.
*  THE 10 BUFFERS MUST BE PRECEDED BY A CALL OF PREDATA, AND
*  THE 10 BUFFERS MUST BE OF THE FORM ... (INPUT,TAPES,TAPE#,...)
*  MONDATA SAVES A MASTER SET OF DATA-CARDS ON TAPE#... THE MASTER
*  SET IS THAT SET OF DATA-CARDS WHICH MONDATA FINDS ON TAPES
*  THE FIRST TIME IT IS CALLED. (A NEW MASTER SET CAN BE CREATED
*  BY USE OF $SAVE, BELOW)
*  ON SUBSEQUENT CALLS, MONDATA READS EDITING REQUESTS
*  FROM INPUT AND PERFORMS THE EDIT, COPYING FROM TAPE# WITH
*  CHANGES TO TAPES. EDITING IS FINISHED WHEN MONDATA FINDS AN
*  EOF OR ANOTHER MONDATA CARD. THEN MONDATA COPIES FROM TAPES
*  TO TAPE# FOR PRINTING (UNLESS SUPPRESSED BY $NOLIST) PLUS TO
*  TAPES IF $SAVE) AND TO PUNCH (IF $PUNCH). FILES ARE PROCESSED
*  SEQUENTIALLY, THUS EDIT REQUESTS MUST APPEAR IN THEIR NATURAL ORDER.
*  MONDATA ALWAYS PRINTS THE EDITING REQUESTS (AND NOTES SUCCESS
*     OR FAILURE TO COMPLY) AND NORMALLY PRINTS THE UPDATED DATA-CARDS.

*  THE EDIT-REQUESTS ARE ... (STARTING IN CARD COLUMN 1) ...

*  ANY CARD NOT OF A'S BELOW IS (INTERPRETED AS) A DATA-CARD
*     THAT IS TO REPLACE THE FIRST CARD IN THE MASTER SET (SEARCH-
*     ING FORWARD ONLY) WHICH MATCHES IT IN COLUMNS 1-20.
*  IN THE FOLLOWING REQUESTS, THE FOLLOWING CONVENTIONS APPLY...)
*     +SIGN MEANS INSERT FROM INPUT ONTO TAPES)
*     -SIGN MEANS SKIP CARDS ON TAPE#)
*     $TRANS MEANS TRANSFER FROM TAPE# TO TAPES)
*     $UPTO MEANS SHIFT-THRU-EXCLUDING THE CARD THAT MATCHES IN COLUMNS
*        1-20 OF THE CARD AFTER THE $UPTO CARD ON INPUT)
*     $INCL MEANS SHIFT-AND-INCLUDING THE CARD THAT MATCHES IN COLUMNS
*        1-20 OF THE CARD AFTER THE $INCL CARD ON INPUT)
*     # REPRESENTS AN INTEGER, WITHOUT LEADING BLANKS, ANY NUMBER
*        OF DIGITS - NOW ENDING ON THE FIRST NON-NUMERIC CHARACTER)
*  +#  MEANS TRANSFER #CARDS FROM TAPE# TO TAPES, UP TO THE MATCHCARD
*  +#! MEANS TRANSFER #CARDS FROM TAPE# TO TAPES, INCLUDING THE
*        MATCHCARD
*  +$  MEANS TRANSFER ALL CARDS FROM TAPE# TO TAPES
*  -#  MEANS INSERT #CARDS FROM INPUT ONTO TAPES, UP TO THE MATCHCARD
*  -#! MEANS INSERT #CARDS FROM INPUT ONTO TAPES, INCLUDING THE
*        MATCHCARD
*  -$  MEANS INSERT THE FOLLOWING ALL CARDS FROM INPUT ONTO TAPES
*  +#  MEANS SKIP #CARDS ON TAPE#, UP TO THE MATCHCARD
*  +#! MEANS SKIP #CARDS ON TAPE#, INCLUDING THE MATCHCARD
*  +$  MEANS SKIP ALL CARDS ON TAPE#

*  $SAVE MEANS USE AS THE NEW MASTER SET, THE SET OF DATA-CARDS
*     YOU HAVE AT THE END OF THE CURRENT EDIT.
```

```
C          $PUNCH MEANS PUNCH THE SET OF DATA-CARDS AT THE END OF THE
C             CURRENT EDIT (REMEMBER TO PROVIDE A PUNCH BUFFER).
C          $* LIST SUPPRESSES PRINTING DATA-CARDS, UNTIL A $LIST CARD
C          $LIST RE-INSTATES  PRINTING DATA-CARDS.
C          $BYPASS SUPPRESSES EXECUTION OF THE CURRENT DATA-SET, UNTIL A
C                  $EXECUTE CARD - IE, MORDATA PROCEEDS TO NEXT EDITING PASS
C                  RATHER THAN RETURNING.  (THIS CARD MAY APPEAR BEFORE FIRST
C                  MORDATA CARD)
C          $EXECUTE RE-INSTATES EXECUTION.
C          MORDATA TERMINATES EDITING PASS (ALTERNATIVE SPELLINGS ARE
C             MORELATA, MORE DATA, AND 7/8/9 ENDOFRECORD).
C
C     SUBROUTINE MORSAV                                                  69.0709
000002      COMMON /MORSAV/ KALL, LIST, LAEV
C     ENTRY MORSAV                                                       69.0709
C                                                                       69/15/69
C
000002      DIMENSION LTC(3), IMAGE(8), KNTRL(8)
000002      DATA (NCH=1),(MASK=777777000000000000008),(LXEV=1)
000002      DATA(LAMBE=99999), (KALD=0), (LIN=5,INPUT),(LPR=6)
000002      DATA (AMP = 71B),(LDT=5),(LPT=6),(LSC=4),(LIST=6)
000002      DIMENSION MESAGE(17)
000002      DATA (MESAGE =
           1   1HM$PUNCH     ,
           2   1HM$SAVE      ,
           3   1HM$NOLIST    ,
           4   1HM$LIST      ,
           5   1HM$BYPASS    ,
           6   1HM$EXECUTE   ,
           7   1HM$UPTO      ,
           8   1HM$INCL      ,
           9   1HM$NUMBER    ,
           0   1HM$*UPTO     ,
           1   1HM$*INCL     ,
           2   1HM$*NUMBER   ,
           3   1HM$-UPTO     ,
           4   1HM$-INCL     ,
           5   1HM$-NUMBER   ,
           6   1HMATCH 1=/,
           7   1HEND.       )
C
C
000002      LPR = LIN
000004      MASK = 0
000004      LTC(1) =LDT $  LTC(2) =LDT  $  LTC(3) = 0
000011      LXEV=
000012      KALD = LAMBE
000014      ASSIGN 1?2 TO NUMBAR
000016      GO TO BUF
000018  2   REWIND LDT
000020      GO TO 900
C
C=====C=C=======*********************************************
C
000021      ENTRY MORDATA
000038      KAL = KALL + 1
000040      IF (KALL - 2) 202,210,214
C ------------------------------------ FIRST CALL
```

```
000032      2?2 LPR = LUT
000033          LTO(2) =LSC $  LTO(2) = LTO(3) = 0
000037          NUM = LAMDA
000041          ASSIGN 216 TO RUMBAR
000042          GO TO 800
000042      2?4 REWIND LSC  $  REWIND LUT
000046          LPR = LSC
000047          LTO(1) = LUT  $  LTO(2)=LTO(3)=0
000053          NUM = LAMDA
000055          ASSIGN 216 TO RUMBAR
000056          GO TO 800
000056      216 REWIND LSC  $  REWIND LUT
000062          IF(LXEG.EQ.0) GO TO 220
000063          GO TO 901
            C
            C ------------------------------------ SECOND CALL
000064      220 READ(LIN,1)  ANAME
000072          IF(ANAME.EQ.2H  )2?4,222
000076      222 IF (ANTHL.NE.7HMONDATA .AND.
           *        ANTHL.NE.4HMONEDATA.AND.
           *        ANTHL.NE.4HMONERR  ATA) GO TO 216
            C
            C ------------------------------------ LATER CALLS
000112      224 READ(LIN,1) ANTHL
000116          IF(ANAME.EQ.2H  )2?4,224
000121      226 CALL READ(0)  $  CALL EXIT
000124      228 REWIND LSC  $  REWIND LUT
000130          LSERR = LPRINC = LYN = MACMENR = 0
            C
            C ------------------------------------ CONSIDER A REQUEST
000136      300 IF (LYN.EQ.0) WRITE (LPR,5)
000141        5 FORMAT(  *1IMAGES OF MONDATA EDIT-CONTROL CARDS*//)
000145          LYN = LYN + 1
000147          IF (EOF,LIN) 301,302
000152          ANTHL(1) = 1HEND RECORD
000156          CALL XMIT(-7,1H ,ANTHL(2))
000162      302 WRITE(LPR,6) ANTHL
000167        6 FORMAT(1H ,A10)
            C
            C ------------------------------------ WHAT KIND OF REQUEST IS THIS
000173          IF (ANTHL.EQ.10HEND RECORD .OR.
           *        ANTHL.EQ.7HMONDATA .OR.
           *        ANTHL.EQ.4HMONEDATA .OR.
           *        ANTHL.EQ.4HMONERR  ATA)  GO TO 400
000237          NUM=LAMDA
000241          NAME=ANTHL.AND.MASK
000245          IF (NAME.EQ.NAME) GO TO 310
000252          NAME=ANTHL.AND.MASK
000256          IF(NAME.EQ.NAM) GO TO 310
000263      308 NAME=
000265          CALL MOMAM(1,ANTHL,NCPW,NAME,1)
000273          IF(NAME.EQ.1M$) GO TO 304
000277          NAME=1A
000300          GO TO 320
            C
000302      310 NPE=2
000303          NAME=3
```

```
000224          CALL *CHAR(2,KNTRL,NCHR,KAR,1)
000230          IF(KAR.EQ.LF) NMS=12
000234          IF(KAR.EQ.CR) NMS=15
000240          IF(KAR.EQ.ESC) NMS=3
000241          N=N+1
000242          GO TO 302
000246      C   CALL *CHAR(1,KR,KNTRL,NCHR,KAR,1)
000252          IF(KAR.EQ.LF .OR. KAR.GT.180) GO TO 306
000256          NAME(N)=NCHR*(KAR-180)
000262          KR=KR+1
000263          GO TO 302

000263      C   IF(N.GT.1) GO TO 310
000266          WRITE(LPR,309)
000271      309 FORMAT(1X,' *UNRECOGNIZED REQUEST*)
000275          MACHERR=1
000276          GO TO 700

            C   ------------------------------------ PROCESS REQUEST

000277      C   310 IF(NMS.GT.0) GO TO 320

            C   ------------------------------------ STATUS-SETTING REQUESTS

000277          GO TO (311,312,313,314,315,316),NMS
000310      311 LPR=SCR=SLEW/ECH   $  GO TO 318
000312      312 LSAVE=LSC          $  GO TO 318
000314      313 LIST=0             $  GO TO 318
000316      314 LIST=LPT           $  GO TO 318
000320      315 LABS=0             $  GO TO 318
000322      316 LABS=1
000323          WRITE(LPR,321) MESAGE(NMS)
000332      321 FORMAT(1X,' *OK *A10)
000332          GO TO 700

            C   ------------------------------------ DATA-MANIPULATING REQUESTS

000333      320 LPR=LSC
000334          LTO(1)=LOT  $  LTO(2)=LTO(3)=0
000340          IF(NMS.GE.10 .AND. NMS.LE.12) LFR=LIN
000354          IF(NMS.GE.13 .AND. NMS.LE.15) LTO(1)=0
000366          IF(NMS.LT.16) WRITE(LPR,321) MESAGE(NMS)
000377          M=(J=2,N)-(NMS+4)
000406          IF(M-3.EQ.0 .OR. NMS.GT.15) GO TO 324
000413          READ(LIT,1) KNTRL
000420          WRITE(LPR,1) KNTRL
000421      324 ASSIGN 326 TO RCMBAK
000427          GO TO RCC
000430      326 IF(MC.EQ.0) GO TO 700
000431          IF(MC.EQ.99) GO TO 330
000432          WRITE(LPR,7)
000436        7 FORMAT(1X,' *FAILURE TO MATCH*)
000443          MACHERR=1
000444          GO TO 700

000445      C   330 IF(NMS.EQ.7 .OR. NMS.EQ.13) BACKSPACE LSC
000452          IF(NMS.EQ.8 .OR. NMS.EQ.11) WRITE(LOT,1) IMAGE
000467          IF(NMS.EQ.16)               WRITE(LOT,1) KNTRL
000477          WRITE(LPR,321) MESAGE(16)
000506          GO TO 700

            C
```

```
C ------------------------------------ PROCESS *END*
000548      400 LFR = LSC
000547          LTO(1) = LIST $ LTO(2) = LTO(3) = 0
000517          NUM = LANGE
000515          WRITE(LPR,321) MESSAGE(17)
000522          ASSIGN 402 TO RETURN
000523          GO TO 800
000524      402 REWIND LSC  $  REWIND LDT
000530          LFR = LDT
000531          LTO(1)=LIST $ LTO(2) = LSAVE  $  LTO(3) = LPUNC
00053A          IF(IMATCH.EQ.0) GO TO 403
000537          LTO(1)=LPT  $  LTO(2)=LTO(3)=0
000543      403 NUM = LANGE
000544          LYES=0
000546          ASSIGN 404 TO RETURN
000547          GO TO 800
000547      404 REWIND LDT
000551          IF(LPR.EQ.0  GO TO 470
000552          IF(IMATCH.EQ.2) GO TO 466
000553          IF(LSAVE.EQ.0) GO TO 480
000554          CALL IMATERR(33H- FAILURE TO MATCH CAN BE FATAL -)
000556      466 CALL PEAK(2)
000560          GO TO 900
C
000561      470 WRITE(LPR,471)
000565      471 FORMAT(/0* 33A -- EXECUTION BYPASSED --*)
000565          GO TO 20
C
000566      480 WRITE(LPR,481)
000572      481 FORMAT(/0* 21A
     *              -- FAILURE TO MATCH CAUSED EXECUTION TO BE BYPASSED --*)
000572          GO TO 200
C
C
C ------------------------------------ READ NEXT REQUEST
000573      700 READ(LIN,1) ANTHL
000601          GO TO 300
C
C ------------------------------------------------------------
C
C         THIS SUBROUTINE USES AS INPUT . . .
C         LFR= LTO(1)=3, NUM(NEG FOR MATCHING), ANTHL(1-2)
C         . . . AS OUTPUT, MESSAGE NUM.
C
000602      800 READ(LFR,1) IMAGE
000610        1 FORMAT(80A1)
000610          IF (EOF,LFR) RVU=810
000613      810   IF(IMAGE.EQ.7H$COMDATA    .OR.
     *            IMAGE.EQ.8H$COMDATA   .OR.
     *            IMAGE.EQ.9H$COMDATA ) GO TO RVU
000624          IF(IMAGE.EQ.7H$BYPASS) GO TO 880
000630          NUM = NUM - ISIGN(1,NUM)
000633          IF (NUM.GE.0) GO TO 820
000634          IF (IMAGE(1).EQ.ANTHL(1).AND.IMAGE(2).EQ.ANTHL(2)) NUM = 0
000644          IF(N.EQ.2) GO TO 890
000645      820 DO 830 L=1,3
000647          LEN = LTO(L)
000651          IF (LEN.LE.0) GO TO 830
```

```
000653          IF ( MESSAGE(5) ) GO TO 880
000654          WRITE(LUN,321) IMAGE
000662    830   CONTINUE
000664          IF(MESSAGE(5)) GO TO 890
000665          GO TO 840
C
000666    840   IF (MOD(LYN,60).EQ.0) WRITE(LUN,2)
000676        2 FORMAT(' ',          IMAGES OF DATA=CAMDS')
000676          IF (MOD(LYN,10).EQ.0) WRITE(LUN,3) (K,K = 1,MO,10),(ANO,K=1,8)
000720        3 FORMAT (1H0,10XI10/1H+,6XGHCAMD  MI,8(4M.........MI))
000720          LYN = LYN + 1
000722          WRITE(LUN,4) LYN, IMAGE
000731        4 FORMAT(1I1,2X80A1)
000731          GO TO 830
C
000732    880   CONTINUE
000733          WRITE(LPR,321) MESAGE(5)
000741          GO TO 890
C
000742    890   GO TO KOMMAN
C     ------------------------------------------------------------
C
C
000744          GO RETURN
C
000747          END  PRECATA
```

```
                      SUBROUTINE PROJ (A,B,C)
                *     SOURCE DATE 68.0209        CONVERT TO CDC 6600
                C     SOURCE DATE 66.0101    BRAND NEW CODE
                C
                C     RETURNS VECTOR C = THE PROJECTION OF A ONTO B DIRECTION
                C
000005                DIMENSION A(3),B(3),C(3)
                C
000005                BB=B(1)**2+B(2)**2+B(3)**2
000010                AB=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
000017                AB=FDIV(AB/BB)                                          68.0209
000023                C(1)=AB*B(1)
000027                C(2)=AB*B(2)
000031                C(3)=AB*B(3)
000032                RETURN
000033                END  PROJ
```

```
      SUBROUTINE QAERROR(N,MESG)
C     SOURCE DATE 68.0209   BRAND NEW CODE
C
C     WRITES ERROR MESSAGE AND TRACES CALLING PROGRAMS
*         ARGUMENT =N= IS IGNORED
*         ARGUMENT =MESG= IS A BCD STRING ENDING ON A DOT OR A DASH OR
*                               THE 100TH CHARACTER
*
C
000004      DIMENSION MESG(1),NAMES(20),LOCS(20)
000004      DATA (WORD=1HMCALLED BY )
C
000004      KAR=0
000005      DO 12 K=11,90
000006      CALL MCHAR(K,MESG,10,KAR,1)
000011      IF(KAR.EQ.1R-.OR.KAR.EQ.1R.) GO TO 14
000022   12 CONTINUE
000024   14 NWD=(K+91)/10
000030      PRINT 1,(MESG(M),M=1,NWD)
000044    1 FORMAT(5H-*** 10A10)
C
000044      DO 24 LEV=1,20
000047      LOC=LEV
000050      NAMES(LEV)=KALLER(LOC)
000055      LOCS(LEV)=LOC
000057      IF(LEV.EQ.1) GO TO 26
000061      DO 22 L=2,LEV
000063      IF(NAMES(L-1).EQ.NAMES(LEV)) GO TO 26
000067   22 CONTINUE
000071   24 CONTINUE
000073      LEV=21
000074   26 LEV=LEV-1
000076      PRINT 2,NAMES(1),LOCS(1),(WORD,NAMES(L),LOCS(L),L=2,LEV)
000121    2 FORMAT(5H0*** 18HERROR DETECTED IN A7 ,6H PLUS O6/
     1  19(13X   A10,      A7 ,6H PLUS O6/))
C
000121      RETURN
000122      END  QAERROR
```

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE RADAR(TARGET,SITE,SIGMA,TRACK,NUMBER)                  1974
*     SOURCE DATE 68.0209       CONVERT TO CDC 6600
C     SOURCE DATE 67.1129   CALL TRNSFM WITH NOERIV = 1
C     SOURCE DATE 67.0704   REPAIR TRNSFM-ING FOR KOORD=2
C     SOURCE DATE 67.0626   DON-T CALL RNV IF SIGMA=0.
C     SOURCE DATE 63.1001
C                                                                       1975
C         TARGET -  STATE ARRAY OF OBSERVED OBJECTS                     1976
C         SITE   -  STATE VECTOR OF RADAR LOCATION                      1977
C         SIGMA  -  ERRORS IN RADAR STATE MEASUREMENT (POLAR COORDINATES)
C         TRACK  -  STATE ARRAY OF RADAR MEASUREMENTS                   1979
C         NUMBER -  NUMBER OF STATE VECTORS IN TARGET AND TRACK ARRAYS  1980
C                                                                       1981
C                   TRACK COORDINATES ARE AS FOLLOWS ...
C
C         ... NUMBER POSITIVE
C            .. TRACK IS RETURNED IN X-Y-Z COORDS. AS COMPLETE STATE
C               VECTOR (NOT RELATIVE TO RADAR)
C
C         ... NUMBER NEGATIVE
C            .. TRACK IS RETURNED IN RADAR POLAR COORDINATES, RELATIVE
C                     TO RADAR LOCATION. DEFINITIONS DEPEND ON KOORD
C               . KOORD = 0 OR 1 -   AZIM IS MEASURED FROM X-AXIS
C                                    TOWARD Y-AXIS
C                                    ELEV IS MEASURED FROM XY-PLANE
C                                    TOWARD Z-AXIS
C               . KOORD = 2      -   AZIM IS MEASURED FROM EAST
C                                    TOWARD NORTH
C                                    ELEV IS MEASURED UP FROM LOCAL
C                                    HORIZON
C                                                                       1985
C         WRITTEN 10/63                                               1986
C                                                                       1987
C     COMPASS,RASCON
000006      COMMON /RASCON/
           1   PROGRM,   KPAGE,    LINE,     TOFDAY,   RUN,     RUNID(6),
           2   MSR,      FLAG,     DATE,     MAXTM,    MAXPG,   MAXLN,
           3   KLASS,    KGROUP,   KUNITS,   KSTINT,   KOORD,   IFEOF
           4  ,NOFILE
C
000006      DIMENSION TARGET(10,2),SITE(10),SIGMA(10),TRACK(10,2),S1(10),     1990
           1 S2(10),ERROR(10),W(3),T(3,3),TDOT(3,3),AXES(10,3),ZU(3)         1991
000006      EQUIVALENCE (S1(2),X),(S1(3),Y),(S1(4),Z),(S1(5),XDOT),(S1(6),    1992
           1 YDOT),(S1(7),ZDOT)                                              1993
000006        DATA (AXES = 30(0))
000006        DATA ((Z(I),I=1,3)=0.,0.,1.)                                    1995
C
000006        NS=IABS(NUMBER)                                                 1996
000010      IF (KOORD.NE.2) GO TO 6                                           1997
000012      CALL UNITV(SITE(2),AXES(2,3))                                     1998
000016      CALL CROSS(ZU,AXES(2,3),AXES(2,1))
000024      IF(SITE(2).EQ.0. .AND. SITE(3).EQ.0.) AXES(2,1)=1.                2001
000037      CALL CROSS(AXES(2,3),AXES(2,1),AXES(2,2))                         2002
000046    6 DO 90 NS=1,NSS                                                    2003
000053      CALL XMIT(-10.0,ERROR)                                          67.0626
000055      DO 1 N=1,10                                                       2004
000062      S1(N)=TARGET(N,NS)-SITE(N)                                        2005
000070      S2(N)=0.                                                          2006
000072   10 IF(SIGMA(N).GT.0.) ERROR(N)=RNV(0.,SIGMA(N))                   67.0626
000110      IF (KOORD.EQ.2) CALL TRNSFM(S1,AXES,S1,1,1)                     67.1129
000121      RSQ=X**2+Y**2+Z**2                                                2009
000126      R=SQRT(RSQ)                                                       2010
000130      RPSQ=X**2+Y**2                                                    2011
000132      RP=SQRT(RPSQ)                                                     2012
000135      RDOT=FDIV((X*XDOT+Y*YDOT+Z*ZDOT),R)                             68.0209
000146      RPDOT=FDIV((X*XDOT+Y*YDOT),RP)                                  68.0209
000155      IF (NUMBER.GT.0) GO TO 20                                         2015
000162      S2(2)=R                                                           2016
```

PRECEDING PAGE BLANK-NOT FILMED

```
000163          S2(3)=AZF(X)                                                    2017
000165          S2(4)=ELF(X)                                                    2018
000167          S2(5)=RDOT                                                      2019
000170          S2(6)=FDIV((X*YDOT-Y*XDOT)/RPSQ)                               68,0209
000177          S2(7)=FDIV((-Z*(X*XDOT+Y*YDOT)+RPSQ*ZDOT)/(RSQ*RP))            68,0209
000213          DO 12 N=2,7                                                     2022
000217    12    S2(N)=S2(N)*ERROR(N)                                            2023
000224          GO TO 80                                                        2024
000224    20    T(1,1)=FDIV(X/R)                                               68,0209
000231          T(1,2)=Y                                                        2026
000232          T(1,3)=FDIV(-X*Z/RP)                                           68,0209
000240          T(2,1)=FDIV(Y/R)                                               68,0209
000244          T(2,2)=X                                                        2029
000245          T(2,3)=FDIV(-Y*Z/RP)                                           68,0209
000253          T(3,1)=FDIV(Z/R)                                               68,0209
000257          T(3,2)=0.                                                       2032
000260          T(3,3)=RP                                                       2033
000261          TDOT(1,1)=FDIV(...R*XDOT-X*RDOT)/RSQ)                          68,0209
000270          TDOT(1,2)=-YDOT                                                 2035
000271          TDOT(1,3)=FDIV(-(RP*(X*ZDOT+Z*XDOT)-X*Z*RPDOT)/RPSQ)           68,0209
000303          TDOT(2,1)=FDIV((R*YDOT-Y*RDOT)/RSQ)                            68,0209
000312          TDOT(2,2)=XDOT                                                  2038
000313          TDOT(2,3)=FDIV(-(RP*(Z*YDOT+Y*ZDOT)-Y*Z*RPDOT)/RPSQ)           68,0209
000325          TDOT(3,1)=FDIV((R*ZDOT-Z*RDOT)/RSQ)                            68,0209
000334          TDOT(3,2)=0.                                                    2041
000335          TDOT(3,3)=RPDOT                                                 2042
000336          DO 25 I=1,3                                                     2043
000343          SUM1=0.                                                         2044
000344          SUM2=0.                                                         2045
000345          DO 24 J=1,3                                                     2046
000346          SUM1=SUM1+T(I,J)*ERROR(J+1)                                     2047
000353    24    SUM2=SUM2+TDOT(I,J)*ERROR(J+1)+T(I,J)*ERROR(J+4)                2048
000367          S2(I+1)=SUM1+S1(I+1)                                           67,0804
000372    25    S2(I+4)=SUM2+S1(I+4                                            67,0804
000376          IF(KOORD.EQ.2.AND.NUMBER.GT.0) CALL TRNSFM(S2,AXES,S2,-1,1)    67,1129
000413          IF(NUMBER.LT.0) GO TO 80                                       67,0804
000415          DO 83 K=2,7                                                    67,0804
000416    83    S2(K)=S2(K)+SITE(K)                                            67,0804
000423    80    S2(1)=S1(1)+ERROR(1)                                            2051
000425          DO 81 N=1,7                                                    67,0804
000427    81    TRACK(N,NS)=S2(N)                                               2053
000440    90    CONTINUE                                                        2054
000443    99    RETURN                                                          2057
000444          END (DATAR
```

```
      SUBROUTINE RITEF (NAMX,VALX,NAX)
C     SOURCE DATE 69.0709    REMOVE ENCODE AND FANCY FORMAT
C     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 67.0512      ADD ENTRIES  RITEON, RITEOFF
C
C     DEBUGGING-OUTPUT ROUTINE
C
C             NAMX -    10-CHARACTER NAME FOR THE VALUES -VALX-
C             VALX -     LIST OF DATA TO BE PRINTED
C             NXX -      NUMBER OF VALUES TO BE PRINTED FROM -VALX-
C
C        EACH TIME CALLED, THIS ROUTINE PRINTS ...
C             * THE NUMBER OF TIMES CALLED
C             * THE NUMBER OF TIMES CALLED WITH THIS -NAMX-
C             * NAMX ITSELF
C             * NXX CONVERTED VALUES FROM -VALX- LIST
C
C     VALX IS CONVERTED BY G OR F FORMAT BY ENTRY RITEF
C                          I                  RITEI
C                          O                  RITEO
C                          A                  RITEA
C
C        NORMALLY A LINE IS SKIPPED BEFORE PRINTING -
C            THIS IS SUPPRESSED IF -NXX- IS NEGATIVE, OR IF THIS
C            -NAMX- IS THE SAME AS LAST TIME.
C                                                                  69.0709
      DIMENSION NAMS(50),NUM(50),VALX(NXX)
000036      DATA (NSEQ=0),(NUM=50(0)),(NAMS=50(-0)),(NV=0)                 69.0709
000046      DATA (NOTYES = 0),(NC=1),(JA=0),(LASNAM=-0)                     69.0709
000036      DIMENSION FORM(8,4)                                             69.0709
000  6      DATA FORM =                                                     69.0709
     1              (40H(* SEQ*I4,* REP*I4,A1),*  =  *,                    69.0709
     *              20H5G20.6/(3X5G20.6)) ,                               69.0709
     2              40H(* SEQ*I4,* REP*I4,A1),*  =  *,                    69.0709
     *              20H10I10/(32X10I10))   ,                              69.0709
     3              40H(* SEQ*I4,* REP*I4,A1),*  =  *,                    69.0709
     *              20H5O20/(32X5O20))     ,                              69.0709
     4              40H(* SEQ*I4,* REP*I4,A1),*  =  *,                    69.0709
     *              20H10A10/(32X10A10))    )                             69.0709
                                                                   68.0209
                                                                   68.0209
            ENTRY RITEI                                            68.0209
                                                                   68.0209
            ENTRY RITEO                                            68.0209
                                                                   68.0209
            ENTRY RITEA                                            68.0209
                                                                   69.0709
                                   RETURN                          69.0709
            NK=   $  IF(KIND.EQ.2 .OR. KIND.EQ.4) NK=10
            NSEQ=NSEQ+1
            NX=IABS(NXX)                                           69.0709
            
            
            NAM=NAMX                                               68.0209
            IF(    .     )   GO TO 13
            
            IF (NAMS(   ).EQ. AM) GO TO 14
```

```
000101      [illegible]
000103          IF [illegible] GO TO 202                                    69.0709
000106      [illegible]
000110          NAMS(NVIEWAR
000112      [illegible]NHEP=NUM([illegible])+1

000114      [illegible]                                                     69.0709
000120          IF [illegible].NE.LASNAM .AND. NXA.GT.0) CALL LSKIP(1)
000137          WRITE(A,FORM[illegible])NSEW,NHEP,NAM,(VALX(N),N=1,NX)      69.0709

000144      [illegible]
000144          RETURN
000144          ENTRY WRITEFF  ; NO YES = NU  $  RETURN                     67.0512
000200          ENTRY WRITEJA  ; NO YES = JA  $  RETURN                     67.0512
000217          END  WRITER
```

```
                                                                        2118
                                                                        2119
                                                                        2120
                                                                        2121
                                                                        2122
                                                                        2123
                                                                        2133
```

```
      SUBROUTINE RORB2 (MODE,VALUE,S1,S2,ORBEL)
C     SOURCE DATE 68.1205   USE KONVERG INSTEAD OF DIVERGE
*     SOURCE DATE 68.0209       CONVERT TO CDC 6400
C     SOURCE DATE 66.1215      SET UP S2(1) ALSO
C
C     RETURNS ORBITAL ELEMENTS BETWEEN S1 AND S2, JUST LIKE SUBR ORB2
C     EXCEPT THAT S2 IS TAKEN TO BE ON A ROTATING PLANET.
C     (POSITIONS S1 AND S2 ARE ASSUMED TO BE CONTEMPORARY)
C
C     FOR MODE = 1, VALUE IS FLIGHT-TIME
C     FOR MODE = 2, VALUE IS EXCESS FLIGHT-TIME (ABOVE MIN-ENERGY ORBIT)
C     FOR MODE = 3, VALUE IS SPEED AT S1 ............. INERTIAL
C     FOR MODE = 4, VALUE IS VEL-ELEV ANGLE AT S1 ... INERTIAL
C     FOR MODE = 5, VALUE IS SPEED AT S1 ............. REL TO GROUND
C     FOR MODE = 6, VALUE IS VEL-ELEV ANGLE AT S1 ... REL TO GROUND
C
C
      COMPOOL,CONCON
000010      COMMON /CONCON/
     1   PI,       SRD,      SLV,      SMF,      SKF,      RBODY,
     2   GACC,     GCON,     WBODY,    RHOZRO,   TWOPI,    HAFPI
C
      COMPOOL,CONCON
000010      DIMENSION S1(10),S2(10),ORBEL(10)
000010      DIMENSION TARG(10),SIMP(10),Z(3),UV(3),VMISS(3)
000010      DATA (Z=0.0,1.),(OKERR=1.)
000010      DIMENSION UU(3,3),GNDVEL(3),V(3),S(10),WORK(8)                    68.1205
C
C
000010      MOD2=MODE    $   IF(MODE.GT.4) MOD2=MODE-2
000015      IF(S2(2).EQ.0. .AND. S2(3).EQ.0. .OR. WBODY.EQ.0.) GO TO 700
000026      CALL CROSS(Z,S2(2),UV)
000032      ABSVAL=ABS(VALUE)
000037      VELTRY=7500.
000040      IF(MOD2.EQ.3) VELTRY=ABSVAL
000044      CALL ORB2(MOD2,VALUE,S1,S2,ORBEL)
000046      TIMTRY=S2(1)-S1(1)
000053      CALL SITEP(S2,TIMTRY,SIMP)
000055      VAL=VALUE
000061      IF(MODE.LT.5) GO TO 50
000063      CALL LOCLAX(S1(2),SIMP(2),3,2,UU)
000071      KOUNT=0
000072      CALL CROSS(Z,S1(2),GNDVEL)
000102      CALL VECLIN(WRODY*SQRT(S1(2)**2+S1(3)**2),GNDVEL,0.,X,GNDVEL)
000122      IF(MOD2.EQ.4) GO TO 40
C
000127   30 ELTRY=0.5
000131   32 VC=ABSVAL*COS(ELTRY)
000135      VS=ABSVAL*SIN(ELTRY)
000140      DO 34 K=1,3
000144   34 V(K)=GNDVEL(K)+VC*UU(K,2)+VS*UU(K,3)
000166      VAL=SIGN(XMAG(V),VALUE)
000174      GO TO 50
C
000177   40 VC=COS(VALUE)
000205      VS=SIN(VALUE)
000212      DO 44 K=1,3
000214   44 V(K)=GNDVEL(K)+VELTRY*VC*UU(K,2)+VELTRY*VS*UU(K,3)
000237      VAL=HAFPI-SEPA(V,UU(1,3))
C
000245   50 CALL KONVSET(10,,OKERR,WORK)                                     68.1205
000250   52 CALL ORB2(MOD2,VAL,S1,SIMP,ORBEL)
000256      S2(1)=SIMP(1)
000263      CALL SITEP(S2,SIMP(1)-S1(1),TARG)
000266      CALL SUBVEC(TARG(2),SIMP(2),VMISS)
000273      EAST=UNIT(VMISS,UV)
000276      IF(KONVERG(EAST,TIMTRY,WORK)) 800,54,60                           68.1205
000305   54 CALL SITEP(S2,TIMTRY,SIMP)
000310      GO TO 52
C
000314   60 IF(MODE.LT.   GO TO 900
C
000316      CALL ORBP(3,.1   ,ORBEL,S)
000322      CALL SUBVEC(S(2),GNDVEL,S(5))
000325      KOUNT=KOUNT+1    $   IF(KOUNT.GT.15) GO TO 800
000335      IF(MOD2.EQ.4) GO TO 400
```

PRECEDING PAGE BLANK-NOT FILMED

```
                C
000337            300 ELNOW=HAPPI-SEPA(S1(2),S(5))
000350                IF(ABS(ELNOW-ELTRY).LT..005) GO TO 900
000354                ELTRY=ELNOW
000355                GO TO 32
                C
000355            400 VELNOW=XMAG(S(5))
000360                IF(ABS(VELNOW-VELTRY).LT. 5.) GO TO 900
000367                VELTRY=VELNOW
000371                GO TO 42
                C
000371            700 CALL ORB2(MODP,VALUE,S1,S2,ORBEL)
000373                GO TO 900
                C
000377            800 CALL TRA OERR(35HSUBR RORB2 HAS FAILED TO CONVERGE =)                68.0209
                C
000401            900 RETURN
000402                END  RORB2
```

```
                      SUBROUTINE RV(S,C,CP)                                    2135
                *     SOURCE DATE 68.0209       CONVERT TO CDC 6600
                C     SOURCE DATE 64.0922
                C                                                              2136
                C         COMPUTES ACCELERATION OF SIMPLE RE-ENTRY VEHICLE     2137
                C                                                              2138
                C         S       =  CURRENT STATE VECTOR OF RE-ENTRY VEHICLE  2139
                C         C       =  BALLISTIC PARAMETER OF R/V  IN KGM/SQ METER 2140
                C                                OR IF KUNITS=1  IN LB/SQ FT
                C         CP      =  NORMAL ACCELERATION COMMAND FOR R/V       2141
                C                                                              2142
                C         WRITTEN 9/22/64                                      2143
                C                                                              2144
     000006           DIMENSION G(3),S(10),CP(3)                               2145
                C
     000006           CALL GRAV(S,G)                                           2146
     000007           X=0.5*DENSITY(S)*FDIV(XMAG(S(5))/C)                   68.0209
     000030           DO 2 N=1,3                                               2148
     000032         2 S(N+7)=X*S(N+4)+G(N)+CP(N)                               2149
     000043           RETURN                                                   2150
     000044           END  RV
```

```
                SUBROUTINE RYDIT(INDENT,NTITLE,MODE,DATA)                                2152
              *     SOURCE DATE 68,0209       CONVERT TO CDC 6400
              C     SOURCE DATE 66,0101
              C                                                                          2153
              C        FILLS DATA INTO ZEROS OF A HOLLERITH TITLE                        2154
              C                                                                          2155
              C        INDENT - NUMBER OF INDENTED SPACES BEFORE PRINTING                2156
              C        NTITLE - HOLLERITH TITLE.  UP TO 10 ZEROS WILL BE FILLED WITH     2157
              C                 ONE DATA VALUE.  AT LEAST ONE ZERO MUST PRECEDE A        2158
              C                 DECIMAL POINT, IF ANY.  LAST WORD MUST BE BLANK.         2159
              C        MODE   - MODE OF THE DATA: 1 FOR INTEGERS, 2 FOR FLOATING        2160
              C                 POINT.  IF MODE IS NEGATIVE, SUPPRESSES PRINTOUT.        2161
              C                                                                          2162
                CCOMPK0,BASCON
000006          COMMON /BASCON/
                1   PROGRM,    KPAGE,    LINE,     TODAY,    RUN,      RUNTO(6),
                2   MSG,       FLAG,     DATE,     MAXTM,    MAXPG,    MAXLN,
                3   KLASS,     KGROUP,   KUNITS,   KSTINT,   KODOD,    IEOF
                4   ,NOFILE
              C
000006          DIMENSION NTITLE(14),NWORD(10),MULT(10),NSVE(14),NFMT(2)              68,0209
000006          EQUIVALENCE (AEQ,IEQ)
              C
000006          NLET=0
000007          NDEC=0
000010          NSP0=1000
000011          DO 12 NWD=1,14                                                        68,0209
000012          NSVE(NWD)=NTITLE(NWD)                                                    2168
000015          IF(NTITLE(NWD) .EQ. 1H ) GO TO 20                                        2169
000017          DO 10 LTR=1,10                                                        68,0209
000020          NSP=LTR*10+NWD-10                                                     68,0209
000024          IF(NSP .GT. NSP0+2) GO TO 12                                             2172
000030          NSYM=0  $  CALL MCHAR(LTR,NSVE(NWD),10,NSYM+1)                        68,0209
000035          IF(NSYM.NE.1R0) GO TO 10                                              68,0209
000042          NLET=NLET+1                                                              2176
000043          NWORD(NLET)=NWD                                                          2177
000046          MULT(NLET)=LTR                                                        68,0209
000050          IF(NSP .EQ. NSP0+2 .OR. NDEC .NE. 0) NDEC=NDEC+1                         2179
000060          NSP0=NSP                                                                 2180
000061    10    CONTINUE                                                                 2181
000063    12    CONTINUE                                                                 2182
000065      20  AEQ=DATA
000071          NDATA=IEQ
000072          IF(IABS(MODE) .EQ. 2) NDATA=DATA*10.**NDEC*SIGNF(.5,DATA)                2184
000105          NAD=IABS(NDATA)                                                          2185
000107          DO 40 NL=1,NLET                                                          2186
000111          KWD=NWORD(NL)                                                            2187
000113          MX=10**(NLET-NL)                                                      68,0209
000120          KNUM=1R                                                               68,0209
000122          IF(NDATA .GE. 0) GO TO 30                                                2190
000123          IF(NAD .GE. MX/10 .OR. NL .EQ. NLET-NDEC-1) KNUM=1R-                  68,0209
000141    30    IF(NAD .GE. MX .OR. NL .GE. NLET-NDEC) KNUM=MOD(NAD/MX,10)+1R0        68,0209
000161          IF(KNUM.EQ.1R0) KNUM=1R0                                              68,0209
000164          IF(MODE.LT.0) CALL MCHAR(10,KNUM,MULT(NL),NTITLE(KWD),1)              68,0209
000202      40  CALL MCHAR(10,KNUM,MULT(NL),NSVE(KWD),1)                             68,0209
000221          IF(MODE .LT. 0) RETURN                                                   2196
000223          IND=IABS(INDENT)                                                         2197
000225          NWD=MIN0(NWD,14-IND/10)                                               68,0209
000233          ENCODE(12,1,NFMT) IND,NWD                                             68,0209
000244      1   FORMAT(4H(3X,I3,4HX,7A10))                                            68,0209
000244          IF(LINE+2 .GT. MAXLN) CALL HEAD(2)                                       2201
000254          LINE=LINE+1                                                              2202
000256          WRITE (6,NFMT) (NSVE(N),N=1,NWD)                                      68,0209
000271          RETURN                                                                   2204
000272          END  RYDIT
```

PRECEDING PAGE BLANK-NOT FILMED

```
                                                                                2206
      FUNCTION SEPA(A,B)          REMOVE CALL OF CROSS                          2207
C     SOURCE DATE  69.0709          CONVERT TO CDC 6400                         220R
C     SOURCE DATE 6R.0209         BRAND NEW CODE                                2209
C     SOURCE DATE 66.0101                                                       2210
C
C         RETURNS THE ANGLE SEPARATING VECTORS A AND B                          2211
C                                                                               2212
      DIMENSION A(3),B(3),C(3)                                                  2213
000005                                                                          2214
C                                                                             6R.0209
      AB=XMAG(A)*XMAG(B)                                                        2216
000005 IF (AB.NE.0.) GO TO 2                                                  69.0709
000014 SEPA=0.                                                                69.0709
000015 RETURN                                                                 69.0709
000016 X=FDIV(DOT(A,B)/AB)                                                    69.0709
000017 2 IF (ABS(X).LT..7) GO TO 4                                              2219
000027 C(1)=A(2)*B(3)-A(3)*B(2)                                                 2220
000033 C(2)=A(3)*B(1)-A(1)*B(3)                                               69.0709
000037 C(3)=A(1)*B(2)-A(2)*B(1)                                                 2222
000043 SEPA=ASIN (FDIV(XMAG(C)/AB))
000044 IF (X.LT.0.) SEPA=3.14159265358979323846264 3-SEPA
000055 RETURN
000061 4 SEPA=ACOS (X)
000063 RETURN
000066 END  SEPA
000066
```

```
      SUBROUTINE SETKORD (K)
C     SETKORD DATE 69/09/12      BRAND NEW CODE
C
C     SETKORD IS FOR YOU IF YOU DON'T NEED COMMON STATEMENTS IN YOUR MAIN PROGRAM
C     ADDITIONAL VARIABLES SET UP #BODY# KUNITS# KSTINT
C
      COMMON/KORD/MASSC(
     1   XXXXXX#    #PAGE#     LINE#     TODAY#    RUN#      RUNIC(8)#
     2   MSG#       FLAG#      LATE#     MAATM#    MAAPG#    MAALN#
     3   #BASE#     #SWOOP#    KUNITS#   KSTINT#   KOUND#    IFEOF
     4   ** FILE
      EQUIVALENCE(MASSC(
C
C     K=0 SURFACE OF FLAT EARTH (KOUND=0) -
C     K=1 SURFACE OF ROUND EARTH (KOUND=1) -
C     K=2 CENTER OF ROUND EARTH (KOUND=2) -
C     K=0 MEAS MES LENGTHS IN METERS (KUNITS=0) -
C     K=1 MEAS MES LENGTHS IN FEET (KUNITS=1) -
C     K=0 INTEGRATION METHOD IS RECTANGULAR (KSTINT=0) -
C     K=1 INTEGRATION METHOD IS RUNGE-KUTTA (KSTINT=1) -
      EQUIVALENCE (A(
C
      K=0
      CALL
      WRITE(
      K=1
C
C     THIS ENTRY MAY BE CALLED WITH INTEGER =1,0,-1 OR WITH REAL OMEGA
      ENTRY SETKORD
      IF(
      WRITE(
      GO TO 12
   10 IF(
      WRITE(
C     CALL EQUIV(
      WRITE(
      WRITE(
   12 RETURN
      END
```

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE SETPLT (A,B,C,D)
*     SOURCE DATE AR,0209        CONVERT TO CDC A400
C     SOURCE DATE A7,0710        BRAND NEW CODE
C
C     GENERAL PURPOSE PRINTER-PLOT ROUTINE ... ARGUMENTS ARE DESCRIBED BELOW
C
000000      COMMON /IMAGE/ IMAGE
000000      DIMENSION IMAGE(101,51),                    LARX(2,6),LABEL(2)        68,0209
000000A     EQUIVALENCE (IC(1))
000036      DATA (MAXX=101),(MAXY=51),(ARG=208)                                   68,0209
000006      DATA CONF/.FALSE./                                                    68,0209
C
C
C-----FOR THIS ENTRY, CALL SETPLOT (X=LOW,
C                                   X=HIGH,
C                                   Y=LOW,
C                                   Y=HIGH)
C         THIS IS THE INITIALIZING CALL - THE ARGUMENTS DEFINE THE
C             ENDS OF THE AXES TO BE PLOTTED
C
C     ENTRY SETPLOT
000036      CONF=.FALSE.                                                          68,0209
000037      CALL XMIT(-1,51,1H ,IMAGE)                                            68,0209
000014      CALL XMIT(-1,1,101H+----------+IMAGE)                                 68,0209
000017      CALL MCHAR(1,1,'+',1,IMAGE,1)                                         68,0209
000023      DO 12 I=1,MAXY                                                        68,0209
000027   12 CALL MCHAR(1,I,'|',1,IMAGE(1,I),1)                                    68,0209
000044      DX=(B-A)/100.
000047      DY=(D-C)/(MAXY-1)
000053      XORG=A  $  YORG=C
000055      IF(.NOT.CONF)GO TO 18                                                 68,0209
000056      IF(DX.LT.0.)XORG=XORG+99.*DX
000063      IF(DY.LT.0.)YORG=5.*DX
000070   18 KVAL=1  $  ASSIGN 19 TO KGO                                           68,0209
000073      DO 20 J=1,4
000074      CALL MCHAR(2+5*J,20*(J-1),
000103      DO 19
000103   19 ARG(I,J)=LABEL(I)
000136   20 ARG(2,J)=LABEL(2)
000113      RETURN
C
C-----THIS ENTRY IS ESSENTIAL TO SETPLOT, EXCEPT IT IS USED FOR
C                   PLOTTING X AND Y TO SAME SCALE
C                   (A COMPUTED CIRCLE IS PLOTTED AS A CIRCLE)
C
000124      ENTRY SETSCL
000123   21 CONF=.TRUE.                                                          68,0209
000126      GO TO 10
C
C
C-----FOR THIS ENTRY, CALL GETINC (DX,
C                                  DY)
C           USE THIS ENTRY IF YOU NEED TO KNOW THE X- AND Y-INCREMENTS
C               WHICH HAVE BEEN COMPUTED BY ISOPLOT
C
000125      ENTRY GETINC
000135   22 A=DX  $  B=DY
000142      RETURN
C
C
C-----FOR THIS ENTRY, CALL PLOTPNT (X=COORD,
C                                   Y=COORD,
C                                   IR  CHARACTER)
C         THIS IS THE ENTRY WHICH ENTERS A POINT IN THE PLOT
C
000142      ENTRY PLOTPNT
000152      KGO=1                                                                 68,0209
000153      GO TO
```

```
000154       24 JX= A+XORG)/DX+1.5
000163          JY=(A+YORG)/DY+1.5
000167          IF(JX.GT.MAXX.OR.JX.LE.0.OR.JY.GT.MAXY.OR.JY.LE.0) RETURN
000210          IF(MGO.EQ.2) GO TO 26                                          68.0209
000212          CALL MCHAR(1A.CI.JX+IMAGE(1.JY).1)                             68.0209
000222          RETURN
C
C
C-------FOR THIS ENTRY, CALL GETPNT (X-COORD,
C                                     Y-COORD,
C                                    IR CHARACTER)
C                    USE THIS ENTRY TO FIND OUT THE CHARACTER
C                    NOW AT (X,Y)
C
000223          ENTRY GETPNT                                                   68.0209
000233          MGO=2                                                          68.0209
000234          C=1.                                                           68.0209
000237          GO TO 24                                                       68.0209
000240       26 CALL MCHAR(JX+IMAGE(1.JY).1).C.1)                              68.0209
000247          RETURN                                                         68.0209
C
C
C-------FOR THIS ENTRY, CALL PLOTXP1 (IR CHARACTER)
C            USE THIS ENTRY TO ADD A CHARACTER TO THE RIGHT OF
C                THE LAST MARK ENTERED
C
000250          ENTRY PLOTXP1
000260          X=JX+1  $  IF(X.GT.MAXX) RETURN
000265          JX=X
000271          CALL MCHAR(1A.CI.JX+IMAGE(1.JY).1)                             68.0209
000301          RETURN
C
C
C-------FOR THIS ENTRY, CALL PLOTYM1 (IR CHARACTER)
C            USE THIS ENTRY TO ADD A CHARACTER BELOW
C                THE LAST MARK ENTERED
C
000302          ENTRY PLOTYM1
000312          Y=JY-1  $  IF(Y.LE.0) RETURN
000316          JY=Y
000322          CALL MCHAR(1A.CI.JX+IMAGE(1.JY).1)                             68.0209
000332          RETURN
C
C
C            YOU MUST CALL PLOTOUT TO GET PLOT PRINTED
C-------FOR THIS ENTRY, CALL PLOTOUT (PRINT-TAPE)
C
000333          ENTRY PLOTOUT
000343          JY=A
000347          IF(IC.LE.0 .OR. IC.GT.A9) IC=A
000362          WRITE(IC.1)
000364       1  FORMAT(1H1)
000364          KVAL=738  $  ASSIGN 14 TO KGO                                  68.0209
000367          DO 14 JYY=1.MAXY  $  JY=MAXY+1-JYY
000375          VAL=YORG+DY*(JY-1)
000402          LABEL(1)=LABEL(2)=1H
000405          IF(MOD(JYY.10).E..1) GO TO 30
000413       14 WRITE(IC.2) LABEL.(IMAGE(I.JY).I=1.1))                         68.0209
000436        2 FORMAT(1A1A10)                                                 68.0209
000436          WRITE(IC.4) (ARG.I=1.6).LABX                                   68.0209
000451        4 FORMAT(20XF1.6(19XF1)/8X12A10/)                                68.0209
000451          RETURN
C
C
000452       30 V=ABS(VAL)
000458          IF(V.LT.1.E-200) GO TO 31
000461          IF(V.GT.9.E+ .OR. V.LT.1.E-2) GO TO 32
000471       31 ENCODE(20.35.LABEL) VAL.KVAL                                   68.0209
000503       35 FORMAT(F17.4.2XR1)                                             68.0209
000503          GO TO 34
000506       32 ENCODE(20.33.LABEL) VAL.KVAL                                   68.0209
000520       33 FORMAT(E17.4.2XR1)                                             68.0209
000520       34 GO TO KGO
C
000526          END  SETPLOT
```

```
                      SUBROUTINE SETSCAL (VALS,NV,VLO,VHI)                                  68,0209
                C     SOURCE DATE 68.0708   MORE CAREFUL TEST BEFORE RETURNING
                *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                C     SOURCE DATE 67.0119       BRAND NEW CODE
                C
                C     CHOOSES AXIS VALUES FOR PLOTS, RETURNING VLO AND VHI, BASED ON
                C     THE NV VALUES IN THE VALS LIST
                C
000006                DIMENSION VALS(NV),IMAGE(2)
                C
000006                VMIN = VALS(INDMIN(NV,VALS))
000015                VMAX = VALS(INDMAX(NV,VALS))
000024                DVREAL = .25*(VMAX-VMIN)
000027                ENCODE(20,1,IMAGE) DVREAL                                             68,0209
000036              1 FORMAT(E20.5)                                                         68,0209
000036                DECODE(10,3,IMAGE) K                                                  68,0209
000050              3 FORMAT(I10)                                                           68,0209
000050                KK = 1
000051                IF(K.EQ.1 .OR. K.EQ.10) KK=2
000062                IF(K.GE.2 .AND. K.LE.4) KK=5
000074                IMAGE(2) = IMAGE(2).AND.77000000000777777778
                     *                    .OR.  33333333300000000B                        68,0209
000101                ENCODE(10,3,IMAGE) KK                                                 68,0209
000111                DECODE(20,1,IMAGE) DVTRY                                              68,0209
000123                IF(K.GE.5) DVTRY = 10.*DVTRY
000130                IF(KK.EQ.5) KK=3                                                      68,0209
                C
000133                MINUS = 1    $   IF(VMIN.GE.0.) MINUS = 0                             68,0209
000137                MODE = 1     $   IF(VMAX*VMIN.LT.0.) GO TO 20
000145                RATIO = ABS(VMAX/VMIN)                                                68,0209
000147                MODE = 2     $   IF(RATIO.LT..5 .OR. RATIO.GT.2.) GO TO 20
000164                MODE = 3
                C
000165              20 GO TO (22,24,22),MODE
                C
000174              22 INT = ABS(VMIN/DVTRY)-MINUS                                          68,0209
000203                 VLO = DVTRY*INT
000210                 GO TO 26
000210              24 VLO = -5.*DVTRY*MINUS
000216              26 VHI = VLO+5.*DVTRY
                C
000225                 IF(VHI.GE.VMAX .AND. VLO.LE.VMIN) RETURN                           68,0708
                C
000234                 GO TO (31,32,33),KK
000243              31 DVTRY = 2.*DVTRY     $   KK = 2    $   GO TO 20
000246              32 DVTRY = 2.5*DVTRY    $   KK = 3    $   GO TO 20                    68,0209
000251              33 DVTRY = 2.*DVTRY     $   KK = 1    $   GO TO 20
                C
000256                 END  SETSCAL
```

```
      SUBROUTINE SITEP (SZ,T,ST)
C     SOURCE DATE  68.0527    TEST FOR .BODY = ZERO
C     SOURCE DATE 66.0922       BRAND NEW CODE
C
C          APPLIES EARTH ROTATION TO STATE SZ FOR TIME T, STORES IN ST
C
      COMPASS,CONCON
000005        COMMON /CONCON/
     1    PI,         SRD,        SLV,        SMF,        SKP,        RBODY,
     2    G  OE,      GCOF,       WBODY,      RHO2RO,     TWOPI,      HAFPI
C
000035        DIMENSION SZ(1  ),ST(10),SS(10)
C
000005        IF(WBODY.EQ.0.0) GO TO 12                                    68.0527
000006        W=T*WBODY
000006        S=SIN(W)
000010        C=COS(W)
000013        DO 10 J=2,6,2
000016        SS(J+2)=SZ(J+2)
000020        SS(J)=SZ(J)*C - SZ(J+1)*S
000025     10 SS(J+1)=SZ(J)*S + SZ(J+1)*C
000032        CALL EMIT (9,SS(2),ST(2))
000035     11 ST(1)=SZ(1)*1                                                68.0527
000041        RETURN
C
000041     12 CALL EMIT(9,SZ(2),ST(2))                                     68.0527
000044        GO TO 11                                                     68.0527
000047        END  SITEP
```

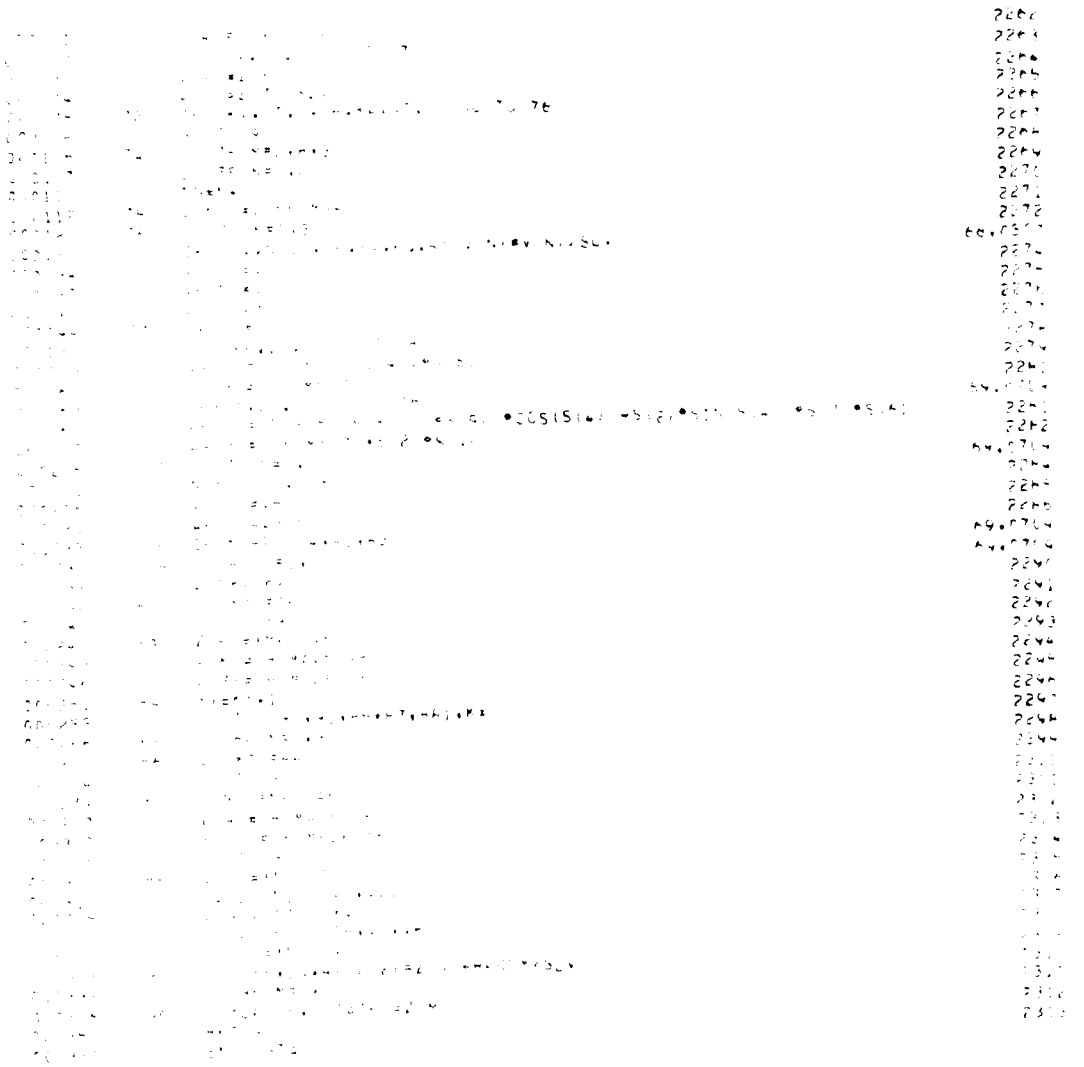
PRECEDING PAGE BLANK-NOT FILMED

```
C     SUBROUTINE SONIC(A,T)
C     SOURCE DATE  AA,0812   CORRECT UNIT-CONVERSION
C     SOURCE DATE  AA,0209       CONVERT TO CDC A400
C     SOURCE DATE  AA,0226   AA AND NEW CODE
C
C     GET MACH NO. AND SPEED AT ALTITUDE ALT (M/SEC AT M, OR FT/SEC AT FT)
C
      COMMON /MASCON/
     1   PROGNM,   KPAGE,    LINE,     TOFDAY,   RUN,      RUNTIM(6),
     2   MSG,      ICLA,     DATE,     MAXTM,    MAXPG,    MAXLN,
     3   * ASS,    ASSIGN,   KUNITS,   KSTINT,   KOORD,    IEEOF
     4  ,NOFILE
C
      COMMON /CONCON/
     1   PI,       S2P,      SLV,      SME,      SKP,      PRODY,
     2   DEGC,     S2PO,     RHOSY,    RHOZRO,   TWOPI,    HAFPI
C
C
      IF (A .
      IF(M.GT.0.3048) GO TO 4                                          6A,0812
      IF(M.GT.0.11350) GO TO 4
      IF(M.GT.0.25000) GO TO 2
      IF(M.GT.0.11110) GO TO 1
      S = 340.-45.*ALOG10(M/0.11110)                                   6A,0209
      GO TO 8
    1 S = 295.
      GO TO 8
    2 IF(M.GT.0.47400) GO TO 3
      S = 295.+62.*ALOG10((M-0.25000)/0.22400)                         6A,0209
      GO TO 8
    3 S = 337.
      GO TO 8
    4 IF(M.GT.0.1500) GO TO 6
      IF(M.GT.0.0000) GO TO 5
      S = 337.-59.*ALOG10((M-0.53350)/0.46650)                         6A,0209
      GO TO 8
    5 S = 314.
      GO TO 8
    6 IF(M.GT.0.0000) GO TO 7
      S = 968.-45.*ALOG10(M-0.01500)/0.01500)                          6A,0209
      GO TO 8
    7 S = SQRT(                                                        6A,0812
      GO TO 8
      END
```

PRECEDING PAGE BLANK-NOT FILMED

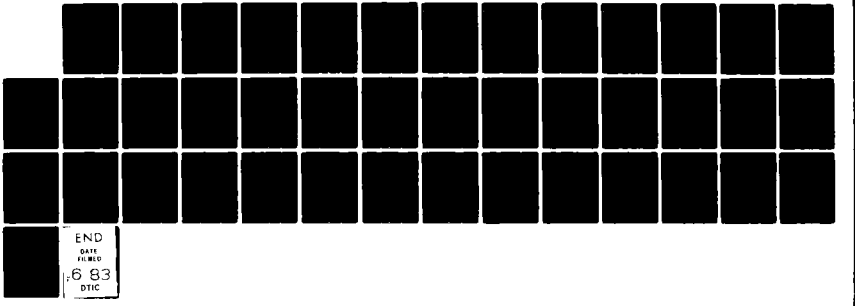
```
      SUBROUTINE STALE(KFORM,STATES,NS,SCALED)                                  2224
C     LAST DATE 04/07/69    MORE CAREFUL CHECK OF USE OF ACCEL COMPONENTS
C     SCRATCH DATE 04/03/67   ALLOW KV = 5
C     SCRATCH DATE 03/03/01
C                                                                               2225
C         SCALES STATE ARRAY FOR OUTPUT                                         2226
C                                                                               2227
C         KFORM  =  OCTIGIT CONTROL WORD FOR SCALING, AS FOR SUBROUTINE         2228
C                   STOUT                                                       2229
C         STATES =  STATE ARRAY TO BE SCALED                                    2230
C         NS     =  NUMBER OF STATE VECTORS FROM STATES TO BE SCALED            2231
C         SCALED =  SCALED VERSION OF STATES                                    2232
C                                                                               2233
C         WRITTEN BY EJ                                                         2234
C                                                                               2235
      COMMON/MASCON/
0000?7    1   KFORM,    KPAGE,    LINE,     TOPDAY,   MCS,      MONTD(6),
          2   YR,       FLAG,     DATE,     MAXTIM,   MAXPG,    MAXLN,
          3   KPASS,    KGMCOM,   KUNITS,   KSTINT,   KOURO,   IFEOF
          4   /S FILE
      COMMON/MASCON/
C                                                                     09/15/69
      COMMON/SCOMCON/
0000?7    1   PX,       SMC,      SLV,      SMF,      SKF,      WBODY,
          2   GMCC,     WCCM,     WRODY,    RMUZMU,   TWOPI,    HAFPI
      COMMON/SCOMCON/
C                                                                     09/15/69
      COMMON/STACON/
0000?7    1   KENT,     MODES(6), NODES(6), CPINS(10),          SETUP(82)
      COMMON/STACON/
C                                                                     09/15/69
0000?7    DIMENSION STATES(10,2),SCALED(10,2),S(10),Z(10),V(3),VR(3),VP(3),     2240
     1 A(3),AM(3),AP(3),VU(3)                                                   2241
0000?7    EQUIVALENCE (S(1),V),(S(R),A(1))                                      224?
0000?7    EQUIVALENCE (MODES(1),MA),(MODES(2),MV),(MODES(3),MP),(MODES(4),      2243
     1 MT),(MODES(5),MS),(MODES(6),MF)                                          2244
0000?7    EQUIVALENCE (NODES(1),NA),(NODES(2),NV),(NODES(3),NP),(NODES(4),      2245
C                                                                               2246
C         KFORM=KFORM                                                           2247
C     1   K=1,6                                                                2248
C         KODES(K)=MOD(K,10)                                                    2249
C         K=K/10                                                                2250
C         IF (K.NE.0) GO TO 2                                                   2251
C         NODES(K)=1                                                            2252
C         NODES(K)=KODES(K)+5                                                   2253
C         KFORM=0                                                               2254
C         DO 4 K=1,NS                                                           2255
C         S(K)=STATES(K,KSTATE)                                                 2256
C     4   Z(K)=S(K)                                                            2257
C         Z(2)=Z(2)/SLV                                                         2258
0000?6    IF (NV.NE.0.OR.NP.NE.0.OR.NF.EQ.5)  Z(5)=Z(5)/SLV                   68.0507
0000?4    IF (NA.GT.1) GO TO 76                                                 2260
0000?0    Z(3)=Z(3)/SLV                                                         2261
```

PRECEDING PAGE BLANK-NOT FILMED

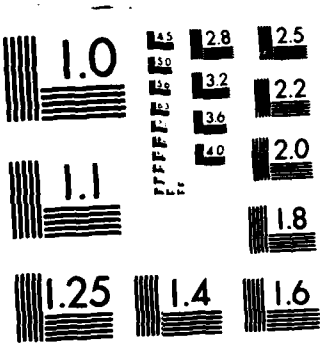


AD-A127 695 THE COMPLEAT TRAIDSMAN(U) GENERAL RESEARCH CORP SANTA BARBARA CA T PLAMBECK 01 SEP 69 GRC-IM-711/2 3/3

UNCLASSIFIED F/G 9/2 NL



END
DATE FILMED
6 83
DTIC



```
      SUBR= TIME,STATE,KFORM,NAMES,STATES,IUNIT=
C     SOURCE DATE AA.0453   OPEN-END READING CAPABILITY
C     SOURCE DATE AA.0204       CONVERT TO FOR A.20
C     SOURCE DATE A7.0724   CALL VLDATA ON READING SPECIAL DATA=A
C     SOURCE DATE A7.0711   REPAIR NEGATIVE
C     SOURCE DATE AA.1222   RE-ARRANGE OUTPUT
C     SOURCE DATE AA.0411   ON KP=4,KV=4 CALL STREP UNLESS DATE
C
C         READS STATE ARRAY, PRINTS IT WITH SUBROUTINE STOUT
C
C         ARGUMENTS ARE DEFINED AS FOR STOUT, EXCEPT THAT -
C              IF KP=1, OUTPUT IS OMITTED
C              IF KN=2, VECTOR NAMES ARE PRINTED BUT NOT READ
C
C         DIGITS OF KFORM ARE  KF,KN,KT,KP,KV,KA
C
C         INPUT STATE VECTORS ARE IN COORDINATE FORMS SPECIFIED FOR THE
C         CARDS, AND MAY REQUIRE EITHER ONE OR TWO CARDS.
C
C         CARD 1 FORMAT IS AS FOLLOWS -
C
C              COLUMNS   FORMAT    CONTENTS
C              -------   ------    --------------------------------
C              1-30      3A8,A4    VECTOR NAME
C              31-40     F10.0     TIME
C              41-70     3F10.0    POSITION (OR VELOCITY) COORDINATES
C              71        I1        POSITION FORMAT KP
C              72        I1        VELOCITY FORMAT KV
C
C         THE ACCEPTABLE P NAMES FOR KP (OR BLANK OR 1=
C         THE ACCEPTABLE PUNCHES FOR KV ARE BLANK OR 1=3
C
C
C         KP AND KV ARE AS DEFINED IN SUBROUTINE STOUT.
C
C         IF DEFINED KP NOR KV IS ZERO, CARD 1 SPECIFIES STATE TIME
C         AND POSITION, AND VELOCITY IS READ FROM A SECOND CARD.
C         IF KP=1 OR KV IS BLANK ON DATA CARD,
C         THESE ARE TAKEN FROM KFORM (SEE STOUT
C
C         CARD 2 FORMAT IS AS FOLLOWS -
C
C              COLUMNS   FORMAT    CONTENTS
C              -------   ------    --------------------------------
C              1-4                 IDENTIFICATION OF TAPE
C              5-70      3F10.0    VELOCITY COMPONENTS
C
C         IF KV IS ZERO, VELOCITY IS ASSUMED TO BE ZERO AND NO CARD 2
C         IS READ.
C
C         IF KP IS ZERO, IT IS RESET TO 1, POSITION COORDINATES ARE SAI
C         IF ZERO, VELOCITY IS TAKEN FROM COLS. 41-70 OF CARD 1
C         ACCORDING TO KV, AND NO CARD 2 IS READ.
C
C         WRITTEN 6/13/68
C
      COMMON /BASIC/
     1   PI,       TWOPI,    CLIGHT,   TDELAY,   MJD,      CENTURY,
     2   MSS,      FLAG,     DATE,     MATTN,    MAXIS,    IAU,
     3   N BSS,    KGMOON,   KUNITS,   KSTINT,   KORBIT,   IEPOS
     4   ,NOFILE
C
      COMMON /CONST/
     1   PI,       SRS,      SLV,      SME,      SMP,      RHOSUN,
     2   SACC,     SCON,     WBODY,    RHOZRO,   TWOPI,    HALFPI
C
      CCOMPR=,STATON
```

```
000006          COMMON /ST&CON/
                1   KFMT,     KMODES(6), MODES(A), CPTNS(10,A),          SRT(P(82)
      C
000006          DIMENSION STATES(10,2),NAMES(5,2),S(10),V(3),VO(3),NAME(5)          2361
000006          LOGICAL NEE TITL                                                   AA.0209
      C
000006          IP=MOD(IABS(KFORM)/100,10)                                         2362
000015          KV=MOD(IABS(KFORM)/10000,10)                                       2363
000024          NEE TITL=IP.NE.0 .AND. LINES.GE.0                                  AA.0209
000032          LAST=IABS(LINES)                                                   2365
000034          DO 25 NSTATE=1,LAST                                                2366
000035          DO 2 I=1,10                                                        2367
000036    2     S(I)=0.                                                            2368
000041          READ(5,3) (NAME(I,J),J=1,4),(S(I),I=1,4),KP,KV                     2369
000065    3     FORMAT(3AH,3A5,4F10.0,2I1)                                         2370
000065          IF(CHEKFIL (5)) RETURN                                             AA.0200
000071          IF(NAME(1).EQ.AMEND DATA) GO TO 30                                 AA.0203
000076          IF(NAME(1).NE.AMOLD DATA) GO TO 1                                  A7.0726
000100          CALL OLDATA(AHSTIN)                                                A7.0726
000101          RETURN                                                             A7.0726
000102    1     IF(NEE TITL) CALL STORE(TITLE,KFORM,NAMES,STATES,0)                AA.0209
000113          NEE TITL=.FALSE.                                                   AA.0209
000114          IF(KP.EQ.0) KP=MOD(IABS(KFORM)/100,10)
000127          KV = KV .AND. 7H
000131          IF ( KV .NE. KV          KV=MOD(IABS(KFORM)/10,10)
      C           IF KV IS BLANK, THAT IS
000144          IF (KP.NE.0) GO TO 5                                               2371
000150          DO 4 I=2,4                                                         2372
000152          S(I+3)=S(I)                                                        2373
000154    4     S(I)=0.                                                            2374
000157          KP=1                                                               2375
000162          GO TO 10                                                           2376
000165    5     IF (KP.GT.1) GO TO 7                                               2377
000165          DO 6 I=2,4                                                         2378
000166    6     S(I)=S(I)*RSL.                                                     2379
000173          GO TO 10                                                           2380
000173    7     S(2)=S(2)*RSLV                                                     2381
000175          S(3)=S(3)/SRD                                                      2382
000176          S(4)=S(4)/SRD                                                      2383
000177          IF (KP.EQ.4) S(2)=S(2)*RHODY                                       2384
000203    10    IF (KV.EQ.0) GO TO 20                                              2385
000206          READ 5,11) (V(I),I=1,3)                                            2386
000216    11    FORMAT(3F10.0)                                                     2387
000216          IF(CHEKFIL (5)) RETURN                                             AA.0209
000222          DO 12 I=1,3                                                        2389
000227    12    V(I)=V(I)                                                          2390
000233          IF (KV.LT.3) GO TO 14                                              2391
000236          V(1)=V(1)*COS(V(2)/SRD)*COS(V(3)/SRD)                              2392
000250          V(2)=V(1)*SIN(V(2)/SRD)*COS(V(3)/SRD)                              2393
000262          V(3)=V(1)*SIN(V(3)/SRD)                                            2394
000267    14    IF (KP.EQ.1) GO TO 16                                              2395
000274          IF (KV.LT.3) GO TO 15                                              2396
000277          V=V                                                                2397
000300          V(3)=V(2)                                                          2398
000301          V(2)=V(1)                                                          2399
000302          V(1)=V                                                             2400
000303    15    IF (KV.LT.2) GO TO 16                                              2401
000306          V(2)=V(2)/(S(2)*COS(S(4)))                                         2402
000312          V(3)=V(3)/S(2)                                                     2403
000314          GO TO 18                                                           2404
000317    16    DO 17 I=2,3                                                        2405
000321    17    V(I)=V(I)/SRD                                                      2406
000326    18    DO 19 I=1,3                                                        2407
000330    19    S(I+4)=V(I)                                                        2408
000334    20    IF(KP.EQ.4) GO TO 202
000336          DO 21 N=1,5
000340    21    STATES(N,NSTATE)=S(N)                                              2409
000351          GO TO 204
000352    212   CALL STREP(STATES(1,NSTATE),5,-1,FDIV(RV/KV))                      AA.0209
000367    204   IF (IP.NE.1) CALL STOUT(TITLE,KFORM,NAME,5,-1)                     2410
000401          IF (KN.NE.1) GO TO 25                                              2411
000406          DO 23 N=1,5                                                        2412
000410    23    NAMES(N,NSTATE)=NAME(N)                                            2413
000420    25    CONTINUE                                                           2414
000423          RETURN                                                             2415
000423       30 LINES=NSTATE-1                                                     AA.0403
000430          RETURN
000430          END  STIN
```

```
                      SUBROUTINE STINT                                                  2417
C                     SUBROUTINE DATE MM/07/7    ZERO OUT SMULS IN DATA STATEMENT
C                     SUBROUTINE DATE MM/DD/YY
C                                                                                       2418
C                     A SPECIAL VERSION OF SUBROUTINE MM, WITH ALL ARGUMENTS IN         2419
C                     COMMON, FOR INTEGRATING SIMULTANEOUSLY SEVERAL TRAJECTORIES.      2420
C                                                                                       2421
C                     ASSUMES INTEGRALS AND THEIR DERIVATIVES ARE CONTAINED IN A        2422
C                     STATE ARRAY LOCATED IN COMMON BEGINNING AT Y, WITH NDIFEQ         2423
C                     STATE VECTORS IN ALL.  INITIAL POSITION AND VELOCITY MUST         2424
C                     BE PROVIDED FOR EACH.  DURING INTEGRATION, FOR EACH STINT         2425
C                     EITHER NEW ACCELERATIONS MUST BE COMPUTED FOR EACH.               2426
C                                                                                       2427
C                      SAVE IS AS FOR MM.                                               2428
C                                                                                       2429
C                      UNLESS KSTINT IS NOT ZERO, SUBSTITUTES RECTANGULAR INTEGRATION   2430
C                     FOR RUNGE-KUTTA INTEGRATION                                       2431
C                                                                                       2432
C                     WRITTEN 6/9/66                                                    2433
C                                                                                       2434
      COMMON/MASCON/
0001          COMMON /MASCON/
              1   PROGMM,    KPASS,     LINE,      TODAY,     RUN,       RUNID(6),
              2   MSG,       FLAG,      DATE,      MAATM,     MAAPG,     MAALN,
              3   KLASS,     KGROUP,    KUNITS,    KSTINT,    KOURO,     IFEOF
              4   /MFILE
      COMMON/MASCON
C                                                                        09/15/69
      COMMON/INTCON
0002          COMMON /INTCON/
              1   DT,        KSTEM,     NDIFEQ,    SOANDX(54)
0002          DIMENSION  S0(12,4),  AINT(12,4)
0002          EQUIVALENCE (S0(1),XINT(1)),SOANDX(7))
      COMMON/INTCON
C                                                                        09/15/69
      COMMON/MULCON
0003          COMMON /MULCON/
              1   SMULS(10,4)
      COMMON/MULCON
C                                                                        09/15/69
0003          DATA (SMULS(N,1),N=1,10)                                     67.0717
0003          1 EQUIVALENCE (SMULS,ST),(SMULS,S1),(XINT,A)
0003          DIMENSION  S1(10),T(10),A(4)

0004          IF (KSTINT.NE.0) GO TO 10                                          2440
              RETURN                                                             2441
              DT2=DT*DT/2.                                                       2442
              DO 4 N=1,NDIFEQ                                                    2443
              DO 4 M=2,4                                                         2444
              S(M-1,N)=S(M,N)+S(M+3,N)*DT+S(M+6,N)*DT2                           2445
0023      4   S(M+3,N)=S(M+3,N)+S(M+6,N)*DT                                      2446
003A          RETURN                                                             2447
003A      1   KSTEP=KSTEP+1                                                      2448
              GO TO (11,13,15,17),KSTEP                                          2449
0050      11  RET                                                                2450
0050          T=T+.5*DT                                                          2451
0053          DO 12 N=1,NDIFEQ                                                   2452
```

```
000[55                  12 M=2,7                                                        2453
000[5A                  S(M=1,N)=S(M,N)                                                 2454
000[5E                  AA=S(M-3,N)*.5                                                  2455
000[71                  A(M-1,N)=AA                                                     2456
000[74              12  S(I,J)=S(M-1,N)+AA*.5                                           2457
000[[                   RETURN                                                          2458
000111              13  DO 14 N=1,NPTS                                                  2459
000113                      14 M=2,7                                                    2460
000116                  AA=S(M-3,N)*.7                                                  2461
000121                  A(M-1,N)=A(M-1,N)+AA*XX                                         2462
000125              14  S(M-1,N)=S(M-1,N)+AA*.5                                         2463
000[[                   RETURN                                                          2464
000[[2              15  END                                                             2465
000[[4                  SUBROUTINE A[1,J)                                               2466
000[[7                  DO 16 M=2,7                                                     2467
000[[7                  AA=S(M-3,N)*.3                                                  2468
000[[6                  A(M-1,N)=A(M-1,N)+AA*XX                                         2469
000[[C              16  S(M-1,N)=S(M-1,N)+AA                                            2470
000[73                  RETURN                                                          2471
000[74              17  DO 18 N=1,NPTS                                                  2472
000[7A                  DO 18 M=2,7                                                     2473
000[7F              18  S(M-1,N)=S(M-1,N)+.16666666666*(A(M-1,N)+S(M+3,N)*(T)          2474
000224                  KSTEP=0                                                         2475
000227                  RETURN                                                          2476
00022[                  END
```

```
      SUBROUTINE STOUT(TITLE,KFLAG,NAMES,STATES,LINES)                    2478
C     KP=KFLAG/100      IF KP#1, CALL SINEM IF IT APPEARS NECESSARY
C     KV=KFLAG-KP*100   ALLOW KV # 5
C     IF(KP.EQ.1) CALL CUTSET WITH 300 IN FIRST ARG
                                                                          2479
C     PRINTS STATE VECTORS WITH TITLE, COLUMN CAPTIONS, NAMES             2480
C                                                                         2481
C     TITLE  -  TITLE PRECEDING ARRAY PRINTOUT                            2482
C     KFLAG  -  OUTPUT FORMAT CONTROL                                     2483
C     NAMES  -  ARRAY OF 5*8HOLD NAMES FOR EACH VECTOR IN STATES          2484
C     STATES -  ARRAY OF STATE VECTORS                                    2485
C     LINES  -  NUMBER OF VECTORS FROM #STATES# TO BE PRINTED             2486
C                                                                         2487
C     KFLAG  CONSISTS OF SIX INDEPENDENT CONTROL DIGITS WHICH ARE,        2488
C     FROM LEFT TO RIGHT, KP, KN, KI, KP, KV, AND KA.  THEIR USE IS       2489
C     AS FOLLOWS -                                                        2490
C                                                                         2491
C     KP    -   FORMAT CONTROL FOR NUMBER PRINTOUT                        2492
C     KP=0        F10.4 CONVERSION
C     KP=1        F15.2 CONVERSION
C                                                                         2495
C     KN    -   NAME OUTPUT CONTROL                                       2496
C     KN=0      -  NAMES NOT PRINTED                                      2497
C     KN=1      -  NAMES PRINTED FOR EACH LINE                            2498
C                                                                         2499
C     KI    -   IDENTIFICATION CONTROL (LEFTMOST PRINTED COLUMN)          2500
C     KI=0      -  NONE                                                   2501
C     KI=1      -  STATE VECTOR ENTRY 1 AS TIME                           2502
C     KI=2      -  STATE VECTOR ENTRY 1 AS MOLLERITH ID                   2503
C     KI=3      -  STATE VECTOR NUMBER IN ARRAY STATES                    2504
C                                                                         2505
C     KP    -   POSITION PRINTOUT CONTROL                                 2506
C     KP=1      -  RECTANGULAR COORDINATES (X,Y,Z)                        2507
C     KP=2      -  POLAR COORDINATES (R,THETA,PHI)                        2508
C     KP=3      -  RADAR COORDINATES (RANGE, AZIMUTH, ELEVATION)          2509
C     KP=4      -  GEOCENTRIC COORDINATES (ALTITUDE, LONGITUDE,           2510
C                  LATITUDE)                                              2511
C     ABOVE VALUE PLUS 5 = POSITIONS LABELLED AS REFERENCE                2512
C                                                                         2513
C     KV    -   VELOCITY PRINTOUT CONTROL                                 2514
C     KV=0      -  VELOCITY PRINTOUT DELETED                              2515
C     KV=1      -  RATES OF POSITION COORDINATES (X DOT, YDOT, ZDOT,      2516
C                  R DOT, THETA DOT, PHI DOT, ETC.)                       2517
C     KV=2      -  RECTANGULAR VELOCITY COMPONENTS (X DOT, YDOT, ZDOT,    2518
C                  R DOT, R * THETA DOT * COS PHI, R * PHI DOT, ETC.)     2519
C     KV=3      -  POLAR VELOCITY COORDINATES (VELOCITY MAGNITUDE,        2520
C                  AZIMUTH, ELEVATION)                                    2521
C     KV=5      -  POLAR COORDINATES OF RADAR HOMESIGHT, STORED IN
C                  VELOCITY COMPONENTS OF STATE.
C     ABOVE VALUES PLUS 5 - STATES(5-7) ARE SCALED AND LABELLED
C                           AS POSITION DEVIATIONS
C                                                                         2524
C     KA    -   ACCELERATION PRINTOUT CONTROL                             2525
C     KA=0      -  ACCELERATION PRINTOUT DELETED                          2526
C     KA=1      -  SECOND DERIVATIVES OF POSITION COORDINATES             2527
C                  (X DOUBLE DOT, Y DOUBLE DOT, Z DOUBLE DOT, ETC.)       2528
C     KA=2      -  RECTANGULAR ACCELERATION COORDINATES (X DOUBLE DOT,    2529
```

```
C                  ETC.; OR - ACCELERATION, THETA ACCELERATION,                  2530
C                  PHI ACCELERATION, ETC.)                                       2531
C          KW=3  -  POLAR ACCELERATION VECTOR (MAGNITUDE, AZIMUTH,               2532
C                  ELEVATION)                                                    2533
C          KW=4  -  ACCELERATION RELATIVE TO VELOCITY (MAGNITUDE,                2534
C                  COMPONENT PARALLEL TO VELOCITY, COMPONENT                     2535
C                  PERPENDICULAR TO VELOCITY)                                    2536
C          ABOVE VALUES PLUS 5 - STATES(8-10) ARE SCALED AND LABELLED
C                               AS VELOCITY DEVIATIONS
C                                                                                2539
C          IF KPRINT IS NEGATIVE, CAPTIONS FOR THE PRINTOUT ARE OMITTED.         2540
C          IF KPRINT IS ZERO, ONLY THE CAPTIONS ARE PRINTED.                     2541
C          IF KPRINT IS POSITIVE, POSITION PRINTOUT IS DELETED                   2542
C                                                                                2543
C          ALL VECTORS IN ARRAY STATES MUST BE IN RECTANGULAR COORDINATES        2544
C          FOR OUTPUT WITH KPRT, AND IN POLAR COORDINATES FOR OUTPUT             2545
C          WITH KW=2, 3, OR 4.                                                   2546
C                                                                                2547
C          IF KFORM IS ZERO OR UNCHANGED FROM ITS VALUE AS STOUT WAS LAST        2548
C          CALLED, EXISTING FORMATS AND HEADINGS IN ARRAY SETUP ARE USED         2549
C          FOR OUTPUT.                                                           2550
C                                                                                2551
C          IF 2 IS ADDED TO KFN ABOVE AND KFORM IS NOT EQUAL TO ITS PRE-         2552
C            VIOUS VALUE, NEW FORMATS ARE INSERTED IN SETUP BUT NO OUTPUT        2553
C          IS PRODUCED.                                                          2554
C                                                                                2555
C          OMITTED B/M/E4                                                        2556
C                                                                                2557
      COMMON/BASCOM/
000010        COMMON /BASCOM/
     1    KPAGHM,    KPAGE,     LINE,      TOFDAY,    MUN,       MONID(6),
     2    MSG,       FLAG,      DATE,      MAATM,     MAAPG,     MAALN,
     3    MPASS,     KSHOLD,    KUNITS,    KSTINT,    KOLRD,     IFEOF
     4   /*/ TILE
C     MAXLEN=
C
C                                                                        09/15/69
      COMMON/CONCOM/
000010        COMMON /CONCOM/
     1    PI,        SK,        SLV,       SMF,       SKP,       KBODY,
     2    DEGR,      DCAN,      KBODY,     RHOZRO,    TWOPI,     HAFPI
      COMMON/STDCOM/
C
C                                                                        09/15/69
      COMMON/STDCOM/
000010        COMMON /STDCOM/
     1    KPRT,      MODES(6), MODES(6), CMINS(108),          SETUP(82)
      COMMON/STDCOM/
C                                                                        09/15/69
000010        DIMENSION NAMES(5,2),STATES(10,2),TCAP(2,3),CAP1(2,9,2),          2562
     1 CAP2(2,3,7),UNITS(2,6,2),Z(15)                                           2563
000010        EQUIVALENCE (MODES(1),KA),(MODES(2),KV),(MODES(3),KP),(MODES(4),   2564
     1 KI),(MODES(5),KA),(MODES(6),KF)                                          2565
000010        EQUIVALENCE (MODES(1),MA),(MODES(2),MV),(MODES(3),MP)              2566
000010        EQUIVALENCE (TCAP(7),CAP1(1)),(CAP1(37),CAP2(1)),(CAP2(43),        2567
     1 UNITS(1)),(TCAP(5),BLANK),(TCAP,CMINS)                                   2568
000010        DATA ITCAP =
     1 12H      TIME      ,12H   SECONDS   ,12H              )                  2570
000010        DATA (CAP1 =
     1 12H          MISSION COORDINATES          ,                              2572
```

```
                2 12H          VELO  ,12HCITY COORDIN ,12HATES          ,                    2573
                3 12H        ACCELE  ,12HRATION COORD ,12HINATES        ,                    2574
                4 12H    REFERENCE   ,12H POSITION CO ,12HORDINATES     ,                    2575
                5 12H    POSITION    ,12HDEVIATION CO ,12HORDINATES     ,                    2576
                6 12H    VELOCITY    ,12HDEVIATION CO ,12HORDINATES     )                    2577
000017          DATA ICAPP =
                1 12H        X       ,12H       Y       ,12H       Z       ,                 2579
                2 12H        M       ,12H     THETA     ,12H      PHI      ,                 2580
                3 12H     RANGE      ,12H   AZIMUTH     ,12H  ELEVATION    ,                 2581
                4 12H  ALTITUDE      ,12H  LONGITUDE    ,12H  LATITUDE     ,                 2582
                5 12H  VERTICAL      ,12H     EAST      ,12H    NORTH      ,                 2583
                6 12H  MAGNITUDE     ,12H   AZIMUTH     ,12H  ELEVATION    ,                 2584
                7 12H  MAGNITUDE     ,12H  PARALLEL     ,12H   NORMAL      ,                 2585
000017          DATA IUNITS =
                1 12H       M        ,12H     M/SEC     ,12H  M/SEC/SEC   ,                 2587
                2 12H      DEG       ,12H   DEG/SEC     ,12H DEG/SEC/SEC  ,                 2588
                3 12H       FT       ,12H    FT/SEC     ,12H FT/SEC/SEC   ,                 2589
                4 12H      DEG       ,12H   DEG/SEC     ,12H DEG/SEC/SEC  )                 2590
000017          DIMENSION MADRS(2,7)                                                        68.0507
000017          DATA(MADRS=12H        RDRES,12HIGHT COORDIN,12HATES                 )       68.0507
000017          EQUIVALENCE (AE,IE)
000017          DATA (KFMT=9999)

000017          IF (KFCHN.EQ.KFMT) GO TO 21                                                 2591
000017          IF (KFCHN.EQ.0) GO TO 21                                                    2592
000017          KFMT=KFCHN                                                                  2593
000017          K=IABS(KFMT)                                                                2594
000017          DO 2 I=1,6                                                                  2595
000017          KODES(I)=MOD(K,10)                                                          2596
000023          K=0                                                                         2597
000025          IF (KODES(I).LE.5) GO TO 2                                                  2598
000031          K=1                                                                         2599
000033          KODES(I)=KODES(I)-5                                                         2600
000035    2     K=K/10                                                                      2601
000042          KUNITS=KUNITS+1                                                             2634
000044          IF(KV.NE.5) GO TO 4                                                        68.0507
000046          CALL XMIT(4,CAP1(1,4,1),MADRS(1,4))                                        68.0507
000054          CALL XMIT(6,MADRS(1,1),CAP1(1,6,1))                                        68.0507
000062          CALL XMIT(2,CAP2(1,1,6),MADRS(1,7))                                        68.0507
000070          CALL XMIT(2,CAP2(1,1,3),CAP2(1,1,6))                                       68.0507
000076    4     KFLAG=2**MOD(KF,2)                                                          2602
000104          INSET=1                                                                     2603
000105          IF (KFCHN.LT.0) INSET=INSET+1                                               2604
000113          IF (KV.EQ.0) INSET=INSET+1                                                  2605
000116          IF (KA.EQ.0) INSET=MIN0(INSET+1,3)                                          2606
000123          IF (KF.NE.0) INSET=4                                                        2607
000125          KU=1                                                                        2608
000126          IF (KP.NE.1) KU=4                                                           2609
000131          KVEL=KP                                                                     2610
000132          KACC=KP                                                                     2611
000133          IF (KP.NE.4) GO TO 12                                                       2612
000135          IF (KV.EQ.2) KVEL=5                                                         2613
000140          IF (KA.EQ.2) KACC=5                                                         2614
000143    12    IF (KV.EQ.3 .OR. KV.EQ.5) KVEL=6                                           68.0507
000154          IF (KA.GE.3) KACC=6+KA/4                                                    2616
000162          KUV=2+KV                                                                    2617
000164          KUA=3+KA                                                                    2618
000166          IF(KV.EQ.3 .OR. KV.EQ.5) KUV=4                                             68.0507
```

```
000176          IF (KA.EQ.3) KUA=6                                              2620
000211          KVV=2-MV                                                        68.0507
000213          IF(KV.EQ.5) KVV=1                                               68.0507
000216          IF (KP.LT.2) GO TO 14                                           2621
000220          IF (KV.EQ.1) KUV=5-MV                                           2622
000223          IF (KA.EQ.1) KUA=6-MA                                           2623
000227    14    MODE=.02*INSET*10                                               2624
000232          IF (KA.NE.0) MODE=MODE+200                                      66.0601
000236          IF (KT.GT.1) GO TO 15                                           2626
000241          IF (KT.EQ.0) MODE=MODE+10                                       2627
000243          CALL OUTSET(MODE,KFN,TCAP(1,1),TCAP(1,2),BLANK,SETUP)           2628
000256          GO TO 16                                                        2629
000257    15    K=9                                                             2630
000261          IF (KI.EQ.3) K=5                                                2631
000264          CALL OUTSET(MODE,K,BLANK,BLANK,BLANK,SETUP)                     2632
000302    16    MODE=(INSET*1)*10+KFN                                           2633
000306          IF (KFUNC.LT.1) GO TO 17                                        2635
000312          CALL OUTSET(MODE,CAP1(1,1,MP+1),CAP2(1,1,KP),UNITS(1,1,KUNITS),  2636
     1    SETUP)                                                                2637
000353          CALL OUTSET(MODE+10,CAP1(1,2,MP+1),CAP2(1,2,KP),UNITS(1,KU,     2638
     1    KUNITS),SETUP)                                                        2639
000435          CALL OUTSET(MODE+20,CAP1(1,3,MP+1),CAP2(1,3,KP),UNITS(1,KU,     2640
     1    KUNITS),SETUP)                                                        2641
000516          MODE=MODE+30                                                    2642
000522    17    IF (KV.EQ.2) GO TO 18                                           2643
000566          CALL OUTSET(MODE,CAP1(1,4,MV+1),CAP2(1,1,KVEL),UNITS(1,KVV,     68.0507
     1    KUNITS),SETUP)                                                        2645
000627          CALL OUTSET(MODE+10,CAP1(1,5,MV+1),CAP2(1,2,KVEL),UNITS(1,KUV,  2646
     1    KUNITS),SETUP)                                                        2647
000732          CALL OUTSET(MODE+20+100,CAP1(1,6,MV+1),CAP2(1,3,KVEL),          2648
     1    UNITS(1,KUV,KUNITS),SETUP)                                            2649
000815          MODE=MODE+30                                                    2650
000822    18    IF (KA.EQ.2) GO TO 19                                           2651
000866          IF (MODE/10.EQ.6) GO TO 19                                      2652
000872          CALL OUTSET(MODE,CAP1(1,7,MA+1),CAP2(1,1,KACC),UNITS(1,3-MA,    2653
     1    KUNITS),SETUP)                                                        2654
001015          CALL OUTSET(MODE+10,CAP1(1,8,MA+1),CAP2(1,2,KACC),UNITS(1,KUA,  2655
     1    KUNITS),SETUP)                                                        2656
001130          CALL OUTSET(MODE+20+100,CAP1(1,9,MA+1),CAP2(1,3,KACC),          2657
     1    UNITS(1,KUA,KUNITS),SETUP)                                            2658
001252    19 IF(KV.NE.5) GO TO 20                                              68.0507
001273          CALL AMIT(4,RALMS(1,4),CAP1(1,4,1))                             68.0507
001341          CALL AMIT(2,RALMS(1,7),CAP2(1,1,6))                             68.0507
001404    20 CONTINUE                                                          68.0507
001410          IF (KP.NE.2) RETURN                                             2660
001414    21    IF (NPTS.GE.1) CALL OUTCOL(TITLE,SETUP,NAMES,STATES,0)          2661
001452          IF (NPTS.EQ.0) RETURN                                           2662
001456          STATES=1.0/STATES                                               2663
001462          DO 30 N=1,NTYPES                                                2664
001473          CALL AMIT(1,STATES(1,N),Z)                                      69.0909
001541          IF(KP.NE.1 .AND. (ABS(Z(3)).GT.TWOPI.OR.ABS(Z(4)).GT.TWOPI))    69.0909
     *                  CALL STREP(Z,Z,1,2)                                   69.0909
001606          CALL STALE(RALMS,Z,1,Z)                                         69.0909
001672          DO 25 K=1,3                                                     2666
001676          IF (KV.EQ.1) Z(K+4)=Z(K+7)                                      2667
001701          IF (KFUNC.EQ.1) GO TO 25                                        2668
001712          Z(K+1)=Z(K+4)                                                   2669
001724          Z(K+4)=Z(K+7)                                                   2670
001735    25    CONTINUE                                                        2671
001737          IEQ=4                                                           
001741          IF (KI.EQ.3) Z(1)=AEG                                           
001754          IF (KI.NE.2) GO TO 30                                           2673
001756          DO 26 K=1,9                                                     2674
001717    26    Z(K)=Z(K+1)                                                     2675
001723    30    CALL OUTCOL(TITLE,SETUP,NAMES(1,N),Z,-1)                        2676
001740          RETURN                                                          2677
001740          END  STOP 1
```

```
                      SUBROUTINE STREP(RSTATE,PSTATE,NSTATE,NDERIV)                    2679
                *     SOURCE DATE 68.0209       CONVERT TO CDC 6400
                C     SOURCE DATE 64.1206
                C                                                                      2680
                C         TRANSFORMS STATE ARRAY FROM RECTANGULAR TO POLAR COORDINATES  2681
                C         OR VICE VERSA                                                2682
                C                                                                      2683
                C         RSTATE -  STATE ARRAY IN RECTANGULAR COORDINATES              2684
                C         PSTATE -  STATE ARRAY IN POLAR COORDINATES                   2685
                C         NSTATE -  NUMBER OF STATE VECTORS IN RSTATE AND PSTATE       2686
                C         NDERIV -  HIGHEST LEVEL OF STATE DERIVATIVES TO BE TRANS-    2687
                C                   FORMED  -                                          2688
                C                             0 - POSITION COORDINATES ONLY TRANSFORMED 2689
                C                             1 - POSITION AND VELOCITY TRANSFORMED     2690
                C                             2 - POSITION, VELOCITY, AND ACCELERATION  2691
                C                                 TRANSFORMED                           2692
                C                                                                      2693
                C         FOR NSTATE POSITIVE, TRANSFORMS FROM RECTANGULAR TO POLAR    2694
                C         COORDINATES.  FOR NSTATE NEGATIVE, TRANSFORMS POLAR TO       2695
                C         RECTANGULAR COORDINATES                                      2696
                C                                                                      2697
                C         WRITTEN 12/6/64                                              2698
                C                                                                      2699
000006                    DIMENSION RSTATE(10,2),PSTATE(10,2),S(10),SP(10)             2700
000006                    EQUIVALENCE (S(2),X),(S(3),Y),(S(4),Z),(S(5),XD),(S(6),YD),  2701
                         1 (S(7),ZD),(S(8),XDD),(S(9),YDD),(S(10),ZDD),(SP(2),R)       2702
                C
000006                    NSS=IABS(NSTATE)                                             2703
000007                    DO 90 NS=1,NSS                                               2704
000011                    IF (NSTATE.LT.0) GO TO 11                                    2705
000012               1    DO 2 N=1,10                                                  2706
000014                    SP(N)=0.                                                     2707
000016               2    S(N)=RSTATE(N,NS)                                            2708
000024                    SP(1)=S(1)                                                   2709
000026                    RSQ=X**2+Y**2+Z**2                                           2710
000032                    R=SQRT(RSQ)                                                  2711
000034                    SP(3)=AZF(X)                                                 2712
000036                    SP(4)=ELF(X)                                                 2713
000040                    IF (NDERIV.EQ.0) GO TO 8                                     2714
000043                    SP(5)=FDIV((X*XD+Y*YD+Z*ZD)/R)                              68.0209
000054                    RPSQ=X**2+Y**2                                               2716
000056                    RP=SQRT(RPSQ)                                                2717
000061                    SP(6)=FDIV((Y*YD+X*XD)/RPSQ)                                68.0209
000073                    SP(7)=FDIV((-Z*(X*XD+Y*YD)+RPSQ*ZD)/(RSQ*RP))               68.0209
000104                    IF (NDERIV.EQ.1) GO TO 8                                     2720
000113                    SP(8)=FDIV((X*XDD+Y*YDD+Z*ZDD+XD**2+YD**2+ZD**2-SP(5)**2)/R) 68.0209
000133                    SP(9)=FDIV((X*YDD-Y*XDD-2.*SP(6)*(X*XD+Y*YD))/RPSQ)          68.0209
000147                    SP(10)=FDIV((-Z*(X*XDD+XD**2+Y*YDD+YD**2)+ZD*(X*XD+Y*YD)+ZDD*RPSQ  68.0209
                         1 -SP(7)*RP*(2.*RP*SP(5)+R*(X*XD+Y*YD)/RP))/(RSQ*RP))       2724
000204               8    DO 9 N=1,10                                                  2725
000210               9    PSTATE(N,NS)=SP(N)                                           2726
000220                    GO TO 90                                                     2727
000220              11    DO 12 N=1,10                                                 2728
000222                    S(N)=0.                                                      2729
000224              12    SP(N)=PSTATE(N,NS)                                           2730
000232                    SNAZ=SIN(SP(3))                                              2731
000235                    CSAZ=COS(SP(3))                                              2732
000240                    SNEL=SIN(SP(4))                                              2733
000243                    CSEL=COS(SP(4))                                              2734
000246                    R=SP(2)                                                      2735
000247                    RD=SP(6)                                                     2736
000251                    S(1)=SP(1)                                                   2737
000252                    X=R*CSEL*CSAZ                                                2738
000255                    Y=R*CSEL*SNAZ                                                2739
000257                    Z=R*SNEL                                                     2740
000261                    IF (NDERIV.EQ.0) GO TO 10                                    2741
000264                    XD=RD*CSEL*CSAZ+R*(-SNEL*CSAZ*SP(7)-CSEL*SNAZ*SP(6))         2742
000275                    YD=RD*CSEL*SNAZ+R*(-SNEL*SNAZ*SP(7)+CSEL*CSAZ*SP(6))         2743
000306                    ZD=RD*SNEL+R*CSEL*SP(7)                                      2744
```

```
000312          IF (NDERIV.EQ.1) GO TO 18                                       2745
000314          XDD= (-SNAZ*CSEL*SP(6)-CSAZ*SNFL*SP(7))*SP(5)                   2746
                1 +(-SNAZ*CSEL*SP(5)-SP(2)*CSAZ*CSEL*SP(6)+SP(2)*SNAZ*SNEL*SP(7))*  2747
                2           SP(6)                                               2748
                3 +(-CSAZ*SNEL*SP(5)+SP(2)*SNAZ*SNEL*SP(6)-SP(2)*CSAZ*CSEL*SP(7))*  2749
                4           SP(7)                                               2750
                5 +CSAZ*CSEL*SP(8)-SP(2)*SNAZ*CSEL*SP(9)-SP(2)*CSAZ*SNEL*SP(10)  2751
000361          YDD= (+CSAZ*CSEL*SP(6)-SNAZ*SNEL*SP(7))*SP(5)                   2752
                1 +(+CSAZ*CSEL*SP(5)-SP(2)*SNAZ*CSEL*SP(6)-SP(2)*CSAZ*SNEL*SP(7))*  2753
                2           SP(6)                                               2754
                3 +(-SNAZ*SNEL*SP(5)-SP(2)*CSAZ*SNEL*SP(6)-SP(2)*SNAZ*CSEL*SP(7))*  2755
                4           SP(7)                                               2756
                5 +SNAZ*CSEL*SP(8)+SP(2)*CSAZ*CSEL*SP(9)-SP(2)*SNAZ*SNFL*SP(10)  2757
000425          ZDD= CSEL*SP(7)*SP(5)+(CSEL*SP(5)-SP(2)*SNFL*SP(7))*SP(7)       2758
                1 +SNFL*SP(8)+SP(2)*CSEL*SP(10)                                 2759
000437     18   DO 19 N=1,10                                                    2760
000441     19   RSTATE(N,NS)=S(N)                                               2761
000451     90   CONTINUE                                                        2762
000454          RETURN                                                          2763
000454          END  STRFP
```

```
      SUBROUTINE SUBHEAD (LEV,HDG)
C     SOURCE DATE  AR.3507   BRAND NEW CODE
C
C     STORES AND PRINTS PAGE-HEADINGS
C
C     IF LEV IS POSITIVE, ENTERS HDG IN (LEV-TH ROW OF) LOCAL ARRAY.
C     IF LEV IS NEGATIVE, DELETES LEV-TH ROW OF LOCAL ARRAY.
C     IF LEV IS ZERO, PRINTS ALL ROWS OF LOCAL ARRAY.
C
CCOMPRG,BASCON
000004      COMMON /BASCON/
     1   PROGRM,    KPAGE,     LINE,      TODAY,     RUN,       RUNID(6),
     2   MSG,       FLAG,      DATE,      MAXTM,     MAXPG,     MAXLN,
     3   KLASS,     KGROUP,    KUNITS,    KSTINT,    KODMD,     IFEOF
     4  ,NOFILE
C
000004      DIMENSION HDG(8),HDGS(8,3)
000004      DATA (HDGS=24(1H ))/(NLV=0)
C
000004      IF(LEV) 2,10,4
C
000005    2 L=-LEV
000010      IF(NLV.EQ.L) NLV=NLV-1
000013      CALL XMIT(-8,1H ,HDGS(1,L))
000020      RETURN
C
000021    4 L=LEV
000024      NLV=MAX0(L,NLV)
000027      CALL XMIT (8,HDG,HDGS(1,L))
000033      RETURN
C
000034   10 KAR=0
000035      IF(NLV.LE.0) RETURN
000040      DO 14 LVL=1,NLV
000043      DO 12 K=11,80
000044      CALL MCHAR (K,HDGS(1,LVL),10,KAR,1) $ IF(KAR.EQ.1H ) GO TO 14
000055      CALL MCHAR (K,HDGS(1,LVL),1,KWD,10) $ IF (KWD.EQ.1H ) GO TO 14
000066   12 CONTINUE
000070      K= 71
000071   14 NWD = (K+9)/10
000075      LINE = LINE + 1
000077   16 WRITE (6,1) (HDGS(N,LVL),N=1,NWD)
000117    1 FORMAT (29X8A10)
000117      RETURN
000121      END  SUBHEAD
```

```
                      SUBROUTINE SUBVEC (A,B,C)
                      SOURCE DATE 66.010    BRAND NEW CODE
                C
                C
                C     RETURNS 3-VECTOR  C =   - B
                C
                C
                      DIMENSION A(3),B(3),C(3)
000005
                C
                      C(1)=A(1)-B(1)
000005                C(2)=A(2)-B(2)
000006                C(3)=A(3)-B(3)
000010                RETURN
000013                END  SUBVEC
000014
```

```
                                                                                                    A5.0004
                                                                                                    A5.0004
```

```
                                                                                                    09/15/69
                                                                                                    69.0526
                                                                                                    69.0526
                                                                                                    69.0526
                                                                                                    69.0526
                                                                                                    69.0526
                                                                                                    69.0205
                                                                                                    69.0526
                                                                                                    69.0526
                                                                                                    69.0526
                                                                                                    69.0205
```

```
      SUBR  TITLER(INDENT,TITLE)                                        2765
C     *REMOVE ENCODE STATEMENT
C     *CONVERT TO CDC 6600
      IMPLICIT INTEGER(A-Z)
                                                                        2766
C     *WRITES CENTERED, INDENTED TITLE                                  2767
                                                                        2768
C     *INDENT - NUMBER OF COLUMNS TO BE INDENTED BEFORE TITLE IS        2769
C               WRITTEN, CENTERED IN REMAINING COLUMNS                  2770
C               (IF NEGATIVE, CENTERING IS OMITTED)                     2771
C     *TITLE  - HOLLERITH TITLE (LAST WORD BLANK, OR LAST               2772
C               CHARACTER A MINUS SIGN)                                 2773
C               (LEAVES BLANK LINE IF FIRST WORD OF TITLE IS BLANK)
                                                                        2774
      COMMON /
     1     NREAD,    LINE,     TOPLAY,   ROW,      MONTH(6),
     2     OUT,      PAGE,     DATE,     MAXTM,    MAXRO,    MAXLN,
     3     LOCK,     ERMSG,    MINITS,   RSTINT,   KOLMD,    IFEOF
      DIMENSION
      EQUIVALENCE
                                                                    09/15/69
      INTEGER   TITLE(10),MB(2)                                       68.0209
      DATA MB/1H ,1H /
                                                                      68.0209
      DATA BLANK/1H /
      DATA IWRITE(6H(1X,A10/1H ) )
     1               123456789A123456789B
C
      NWORD=0                                                         68.0209
      DO 1 I=1,10                                                     68.0209
      IF(TITLE(I).EQ.BLANK) GO TO 10                                  68.0209
    1 NWORD=I                                                         68.0209
      I=1                                                             68.0209
    2 CALL MCHAR(TITLE(I),MB,1)     $  IF(MB.EQ.1H=) GO TO 2          68.0209
      CALL MCHAR(TITLE(I),MB,10)    $  IF(MB.EQ.BLANK) GO TO 2         68.0209
      I=I+1                                                           68.0209
      IF(I.LE.NWORD) GO TO 2                                          68.0209
C
   10 IF(NWORD.EQ.0) GO TO 10                                           2802
      IF(INDENT.LT.0) GO TO 8                                           2803
      NBLANK=(INDENT+KOLUMS)/2                                        68.0209
      DO 3 I=1,NWORD    $  NWORD=MAX0(NWORD,I)                        69.0709
      CALL MCHAR(10,BLANK,I,FMT,1)    $  CALL MCHAR(10,BLANK,7,FMT,1) 69.0709
      CALL MCHAR(10,BLANK,I,FMT,1)    $  CALL MCHAR(10,BLANK,6,FMT,1) 69.0709
      CALL MCHAR(10,BLANK,I,FMT,1)    $  CALL MCHAR(10,BLANK,10,FMT,1)69.0709
    3 CONTINUE
      IF(NBLANK.GT.0) CALL MEAD(1)                                      2809
      WRITE(6,IWRITE) (TITLE(I),I=1,NWORD)                            68.0209
      RETURN                                                            2811
    8 NBLANK=0                                                        68.0209
      GO TO 2                                                         68.0209
      END TITLER
```

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE TITLIN (TITLE)
C     ** HCD DATE 4A.0701   BRAND NEW CODE
C
C        READS HCD CARD IMAGE (8A10) AND RETURNS IN *TITLE*
C
000003      DIMENSION TITLE(8)
C
000003      READ(5,1) TITLE
000012    1 FORMAT(8A10)
000012      IF(CHKFIL(5)) RETURN
000016      NET *4
000017      END
```

```
                      SUBROUTINE TRNSFM(S,AXES,SP,NS,NDERIV)                    2813
C     SOURCE DATE 04/1207
C                                                                               2814
C     TRANSFORMS STATE ARRAY TO OR FROM ROTATED COORDINATE FRAME                2815
C                                                                               2816
C     S      -  STATE ARRAY IN BASIC COORDINATE FRAME                           2817
C     AXES   -  DIRECTION COSINE ARRAY FOR PRIMED COORDINATE FRAME              2818
C     SP     -  STATE ARRAY IN PRIMED COORDINATE FRAME                          2819
C     NS     -  NUMBER OF STATE VECTORS IN STATE ARRAY                          2820
C     NDERIV -  HIGHEST LEVEL OF TIME DERIVATIVE TO BE INCLUDED                 2821
C                                                                               2822
C     FOR NS POSITIVE, TRANSFORMS STATES FROM BASIC TO PRIMED COORD.            2823
C     FOR NS NEGATIVE, TRANSFORMS STATES FROM PRIMED TO BASIC COORD.            2824
C                                                                               2825
C     WRITTEN 12/7/64                                                           2826
C                                                                               2827
000006                DIMENSION S(10,2),AXES(10,3),SP(10,2),T(10,10),SW(10)     2828
000006                DATA ( T = 100(0) )
C
000006                DO 1 I=1,3                                                2830
000010                DO 1 J=2,4                                                2831
000011                T(I+1,J-1)=AXES(J-1,I)                                    2832
000017                T(I+4,J-1)=AXES(J+3,I)                                    2833
000025                T(I+4,J+3)=AXES(J-1,I)                                    2834
000031                T(I+7,J-1)=AXES(J+6,I)                                    2835
000037                T(I+7,J+3)=AXES(J+3,I)*2.                                 2836
000044     1          T(I+7,J+4)=AXES(J-1,I)                                    2837
000054                NSS=IABS(NS)                                              2838
000056                MX=4,3*NDERIV                                             2839
000063                IF (NS.LT.0) GO TO 5                                      2840
000061                DO 4 N=1,NSS                                              2841
000063                SP(1,N)=S(1,N)                                            2842
000070                DO 2 I=2,10                                               2843
000071     2          SW(I)=S(I,N)                                              2844
000102                DO 4 I=2,MX                                               2845
000104                T=0.                                                      2846
000105                DO 3 J=2,MX                                               2847
000107     3          T=T+T(I,J)*SW(J)                                          2848
000121     4          SP(I,N)=T                                                 2849
000131                RETURN                                                    2850
000132     5          DO 8 N=1,NSS                                              2851
000134                S(1,N)=SP(1,N)                                            2852
000141                DO 8 I=2,10                                               2853
000142     8          SW(I)=SP(I,N)                                             2854
000153                DO 8 J=2,MX                                               2855
000155                T=0.                                                      2856
000156                DO 7 I=2,MX                                               2857
000160     7          T=T+T(I,J)*SW(I)                                          2858
000172     8          S(I,N)=T                                                  2859
000232                RETURN                                                    2860
000233                END TRNSFM
```

PRECEDING PAGE BLANK-NOT FILMED

```
                      FUNCTION TRPLATE(A,AS,FS,NAF,NPT,NX,NF)
                      C  PURPOSE  APPLIES A QUADRATIC  WHANG NEW CODE  PATTERNED AFTER FORTRAN/S
                      C
                      C  USAGE  - FLOAT IN-LINE FUNCTION ROUTINE
                      C
                      C  DESCRIPTION OF PARAMETERS  - WHERE A(I) LISTS OF FUNCTION VALUES (FS) AND ABSCISSAS (AS),
                      C  THIS FUNCTION RETURNS A FUNCTION VALUE FOR THE GIVEN ABSCISSA (A) OBTAINED
                      C  BY QUADRATIC INTERPOLATION.
                      C  THE TWO ARRAYS (AS AND FS) EACH CONTAIN NAF+1 VALUES TO BE USED --
                      C  THESE VALUES ARE (XA AND NF RESPECTIVELY) NUMBERS APART IN CORE.
                      C  IF THE A OR AS IS DIMENSIONED (NA,NAF,) AND FS IS DIMENSIONED (NF,NAF)
                      C
                      C
000010                DIMENSION AS(NX,NAF),FS(NF,NAF)
                      C
000012                IF (AS(1,1)-GT.AS(1,NAF)) GO TO 10
                            K = 1
                            KAF
000020                IF (A.LE.AS(1,K)) GO TO 20
000025                GO TO 15
000027                K = KAF
000030                GO TO 20
                      C
000032          15    K = 2 * NAF
000042                IF (K.GT.KAF.AND.A.LT.AS(1,K)) GO TO 20
000048                GO TO 15
000052                K = KAF
                      C
000054          20    KHI = K
000055                KLO = KHI - 1
000057                NFF = NAF - 2
000060                IF (NAF.LE.0) GO TO 50
000062                PHOLD = (AS(1,KLO) - AS(1,KHI)) * (AS(1,KLO) - AS(1,KHI))
000076                KL =
000077                KH =
000078                GO TO
                      C     ARE ALL ARE REDUCED
000101                IF (KL.LT.0) GO TO 32
000103                IF (KLO.LE.1) GO TO 32
000105                IF (ABS(AS(1,KLO-1)) .LT. PHOLD) GO TO 32
000120                KL = KLO - 1
000122          32    IF (KHI.GT.NX) GO TO 34
                      C     ARE ALL KHI ARE RAISED
000124                IF (KHI.EQ.NX) GO TO 34
000126                IF (ABS(AS(1,KHI)) - AS(1,KHI+1)) .LT. PHOLD) GO TO 34
000142                KH = KHI + 1
000144          34    IF (KH - KL + 1.GT.0) GO TO 40
                      C     DETERMINE IF BETTER FIT WHICH PERMITTED
000146                IF (KL.EQ.1) GO TO 37
000150                IF (ABS(A-AS(1,KHI)) .LT. ABS(A-AS(1,KL-1)) GO TO 36
000164          36    KL = KL
000165                KH =
000166                GO TO 40
000170          37    IF (KL.LT.0) GO TO 38
000172          38    KHI = KH
000174                KH =
000176          40    CONTINUE
                      C
000177                TRP = 0.
000200                   SX = AS(1,KH) - AS(1,KHI)
000206                   SA = A - AS(1,KHI)
000214                   SY = FS(1,KH) - FS(1,KHI)
000222                   IF (SX.EQ.0.) GO TO 42
000226                   TRP = FS(1,KHI) + (A - AS(1,KHI))/
                     *                  (AS(1,KH) - AS(1,KHI)) * SY
000245          42    CONTINUE
000246          46    TRP = TRP + PHOLD
000252                TRPLATE = TRP
000253                RETURN
000254                END
```

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE TRPSTA (ST,T,S)
C     SOURCE DATE 68.070   BRAND NEW CODE
C
C     INTERPOLATES FROM TWO STATES *ST*, TO FIND *S* AT TIME *T*
C    ** NOTE ** ALL TEN COMPONENTS OF BOTH *ST* VECTORS MUST BE PROVIDED
C
000005      DIMENSION ST(10,2),S(10)
000005      DIMENSION X(5),U(5),Y(10)
C
000005      CALL XMIT(-9.0,S(2))
000010      SUM=ST(1,1)+ST(1,2)
000014      DO 4 KTM=1,2
000015      CALL XMIT(10.,ST(1,KTM),Y)
000021      SUM=Y+SUM
000024      X=T+Y-SUM
000027      DO 2 J=2,5
000031      U(J)=Y(J-1)**J
000034    2 X(J)=X(J-1)*Y
000041      DO 4 J=2,4
000042      A=(-20.*Y(1)+20.+8.*U*Y(J+3)+   U(2)*Y(J+6))/(2.*U(3))
000054      B=(+30.*Y(1)-30.+14.*U*Y(J+3)-2.*U(2)*Y(J+6))/(2.*U(4))
000070      C=(-12.*Y(1)+12.+ 6.*U*Y(J+3)+   U(2)*Y(J+6))/(2.*U(5))
000102      S(J-1)=S(J-1)+   A*X(3)+    B*X(4)+    C*Y(5)
000111      S(J+3)=S(J+3)+3.*A*X(2)+ 4.*B*X(3)+ 5.*C*X(4)
000124    4 S(J+6)=S(J+6)+6.*A*X(1)+12.*B*X(2)+20.*C*X(3)
000142      S(10)=T
000143      RETURN
000143      END  TRPSTA
```

PRECEDING PAGE BLANK-NOT FILMED

```
C ** TITLE UNAMIT (VARNL
C ** 03/01/70    ADD 2 UNIT-NAMES, REVISE FORMAT IF NAME USED
C ** 02/27/70    MAKE CONVERSIONS CENTRAL OR KUNITS
C ** 02/12/70       CONVERT TO CDC CODE
C ** 01/01/70    CHANGE NEW CODE
C
C     SUBROUTINE, ALLOWING CHANGE OF VARIABLES SELECTED BY NAME
C
C     FOR SUCCESSIVE CALLS OF UNAMIT, ROUTINE DOES THREE TASKS ...
C          READS A NAME AND A VALUE FROM A CARD (SEE FORMAT BELOW)
C          STORES THE VALUE READ IN =VARNL=
C          SAVES THE NAME AND ADDRESS OF =VARNL=
C          (UNAMIT HAS ROOM FOR 30 DISTINCT VARIABLE-NAMES)                    69.0709
C
C     WHEN CALLED WITH BY ENTRY UGETIT, THE PROCEDURE IS ...
C          READ A NAME AND A VALUE FROM A CARD
C          SEARCH LIST OF NAMES AND FIND ADDRESS OF MATCHING VARBL
C          STORE VALUE IN THAT VARBL, AND GO READ THE NEXT CARD
C        S  THE UGETIT STOPS AT THE CARD WHOSE =NAME= IS =END DATA=
C
C     ***  NOTE - VALUES READ FROM CARDS ARE CONVERTED TO INTEGERS
C               IF THE FIRST CHARACTER OF =NAME= IS I THRU N
C
C     CARD COL  DESCRIPTION
C     --------  -----------------
C       1-10       NAME IDENTIFYING VARIABLE                                  69.0709
C       11-20      ANY FURTHER DESCRIPTION OF VARIABLE                        69.0709
C       21-30      VALUE, IN F10
C       31-40      MULTIPLIER, IN F10 (TAKEN AS UNITY IF BLANK)
C       41-50      DIVISOR, IN F10    (TAKEN AS UNITY IF BLANK)
C
C     *** NOTE - OPTIONALLY, MAY PUNCH A UNIT-NAME IN COL 51-53
C               ALLOWABLE UNIT-NAMES ARE LISTED UNDER SUBR INI
C
C
C     TYPICAL USE IS AS FOLLOWS ...
C
C          CALL UNAMIT(A)
C          CALL UNAMIT(B)
C          CALL UNAMIT(C)
C        1 (C  WRITE,PRINT,ETC)
C          CALL UGETIT
C          GO TO 1
C
C     ***********************
      SUBROUTINE UNAMIT (NAME, VALUE, IFLAG)
      COMMON /UNAMI/ NAME, NAMVAL(3,11)                                        69.0709
      DATA NAME/11*1H /                                                        69.0709
      DATA (NAMVAL(I,1),I=1,3)/                                                68.0527
     1   3HEND,   2.17453242527,                                               68.0527
     2   2HET,    0.30448006,  1.0,                                            68.0527
     3   3HFSH,   4.HH024,  1.0,                                               68.0527
     4   2HHH,    1000.,  3280.833,                                            68.0527
     5   3HMI,    1H53.25,  8080.3,                                            68.0527
     6   2HLB,    0.453592,  1.0,                                              68.0527
     7   1H ,     9.80665,  32.17398,                                          68.0527
     8   3HFT,    304.8006,  1000.,                                            68.0527
     9   1HM,     1.0,   3.2H0833 ,                                            69.0709
     C   3HMM,    2(1.0) ,                                                     69.0709
     1   3HSEC,   2(1.)  )                                                     69.0709
C     END,COMMENT
```

PRECEDING PAGE BLANK-NOT FILMED

```
                                                                                09/15/69
                                                                                68.0204
                                                                                68.0209
                                                                                68.0527
                                                                                69.0709
                                                                                68.0204
                                                                                69.0709
                                                                                68.0204
                                                                                68.0204
                                                                                69.0709
                                                                                69.0709
                                                                                68.0527
                                                                                68.0527
                                                                                68.0527
                                                                                69.0709
                                                                                69.0709
                                                                                69.0709
                                                                                68.0204
                                                                                68.0204
                                                                                68.0204
                                                                                68.0527
                                                                                69.0709
```

```
      SUBROUTINE UNITV(A,B)                                             AAA2
C     SOURCE DATE AA/02/24          CONVERT TO C  4/24/
C     SOURCE DATE AA/03/01    RHEAD AFW 03/8
C
C         RETURNS B, A UNIT VECTOR PARALLEL TO VECTOR A                 AAA3
C                                                                       AAA4
      DIMENSION A(3),B(3)                                               AAA5
C                                                                       AAA6
000004      R=SQRT(A(1)*A(1)+A(2)*A(2)+A(3)*A(3))                       AAA7
000016      DO 1 I=1,3                                                  AAA8
000018    1 B(I)=A(I)/R                                                 AA,0203
000032      RETURN                                                      AAA9
000034      END UNITV
```

```
C  SUBROUTINE UPSTATE (ST,T,S)
C  SOURCE DATE 87/07/17   BRAND NEW CODE
C
C  SIMPLE RECTANGULAR METHOD, ADVANCES #STA TO TIME #TA# STORES IN #ST
C
      DIMENSION ST(1),S(1)
C
      T=ST(1)
      HALFDT=DT/2.
      DO 10 K=2,4
      ST(K)=ST(K)+HALFDT*DYDST(K+4)
      ST(K+3)=ST(K+3)+DT*DYDST(K+4)
   10 CONTINUE
      RETURN
      END
```

```
                SUBROUTINE VECLIN (A,AA,B,BB,CC)
            C   SOURCE DATE A6,0101   BRAND NEW CODE
            C
            C   RETURNS  CC(1-3) = A*AA(1-3) + B*BB(1-3)
            C
000006          DIMENSION AA(3),BB(3),CC(3)
            C
000006          DO 10 J=1,3
000010       10 CC(J)=A*AA(J) + B*BB(J)
000017          RETURN
000020          END  VECLIN
```

```
      SUBROUTINE VECSUM (A,AA,B,BB,CC)
      SOURCE DATE AA-1215          BRAND NEW CODE
C     ADDS TO CC(1-3) THE VECTOR A*AA(1-3) + B*BB(1-3)
      DIMENSION AA(3),BB(3),CC(3)
      DO 10 J=1,3
   10 CC(J)=CC(J) + A*AA(J) + B*BB(J)
      RETURN
      END  VECSUM
```

000026
000006
000010
000021
000021

PRECEDING PAGE BLANK-NOT FILMED

```
      SUBROUTINE WHEN
C     SOURCE DATE 08.0209        CONVERT TO CDC 6400
C
C     WRITES TIME/DATE/CALLINGPROGRAM MESSAGE EVERY TIME CALLED
C         INFORMATION PRINTED IS ...
C                   NO= MANY TIMES =WHEN= HAS BEEN CALLED
C                   NAME OF CALLING PROGRAM
C                   REL ADDR OF CALL STATEMENT WITHIN THAT PROGRAM
C                   TIME OF DAY
C                   CP TIME SINCE BEGINNING OF JOB
C                   CP TIME SINCE LAST CALL OF =WHEN=
C                   PP TIME SINCE BEGINNING OF JOB
C                   PP TIME SINCE LAST CALL OF =WHEN=
C
C
000002      DATA (TOLD=0),(KOUNT=0),              (PTOLD=0.)
C
000002      TOD=DTIME(TOD)
000005      CALL SECOND(T)
000006      DT=T-TOLD  $  TOLD=T
000011      CALL PSECOND(PT)
000012      DPT=PT-PTOLD  $  PTOLD=PT
000015      KOUNT=KOUNT+1
000016      N=1
000017      KALD=KALLER(N) ,4HO.  77 77 77 77 77 77 77 000000 B
000023      CALL COUNOUT(1)
000024      WRITE(6,1) KOUNT,KALD,N,TOD,T,DT,PT,DPT
000050    1 FORMAT(/I3,7H/ FROM A1,3H + O6,7H/ TIME A10,10H/ CP TIME F8.3,
     *     *(+F7.3)/ PP TIME F8.3,(+F7.3)* )
000050      RETURN
000051      END  WHEN
```

PRECEDING PAGE BLANK-NOT FILMED

```
                      FUNCTION XMAG(X)                                          2872
                      SOURCE DATE A6,0101   BRAND NEW CODE
                C
                C                                                               2873
                C     RETURNS THE MAGNITUDE OF VECTOR X                         2874
                C                                                               2875
                C                                                               2876
000003                DIMENSION X(3)                                            2877
000003                XMAG=SQRT(X(1)*X(1)+X(2)*X(2)+X(3)*X(3))                  2878
000014                RETURN
000014                END   XMAG
```

```
                      INTEGER FUNCTION XMIXER(I)                                2880
C                     SOURCE DATE 86.0101   BRAND NEW CODE
C                                                                               2881
C                         RETURNS I WITH A FLOATING POINT NAME                  2882
C                                                                               2883
000003                XMIXER=I                                                  2884
000004                RETURN                                                    2885
000005                END  XMIXER
```

5. QUICK REFERENCE

5.1 TYPICAL DECK SETUP

JOB,9[illegible]4-job,priority,cpseconds,fieldlength.	
[illegible]	CØMPKG. RUN,S,,,MERGE0.
[illegible] EXECUTE. ?	
	Commonname fortran common statements ENDCOM commonname fortran common statements ENDCOM ENDALL repeat up to 62 times
[illegible]	(including CCOMPKG,commonname cards)
[illegible]	

5.2 TYPICAL PROGRAM STRUCTURE

```
      PROGRAM name (INPUT, OUTPUT, TAPE6=OUTPUT,
                    TAPE5=INPUT) if not using PREDATA
                    TAPE5) if using PREDATA
                    TAPE5, TAPE4) if using PREDATA and MORDATA
      DIMENSION S(10), C(3,16), CP (13 or more), ORB (10), SETUP (82)
      EXTERNAL  PLAS, FLIER, RV, FOAL

      CALL PREDATA
     1  CALL HEAD (6HPROGRNAME PROGRAMDESCRIPTION-)
      CALL HEAD (1)
      CALL SETPOK (n)
     2  CALL input routines
      output
      CALL output routines

      if you want to read data for another case
      CALL CONTROL (-60)
      GO TO 2

      if you want to start another run (new page heading, etc.)
      GO TO 1

      END
```

5.3 CALLING SEQUENCES

Presented on the next four pages is a table of subroutine calls and function references, showing the standard form for arguments. The routines are grouped into functional categories, and then alphabetized. Definitions of codes and dimensions of arrays are noted (in the least-obvious instances anyway). The most often-used arguments are:

S(10) -- state vector

ORB(10) -- orbital-element vector

(*,16) -- rocket characteristics

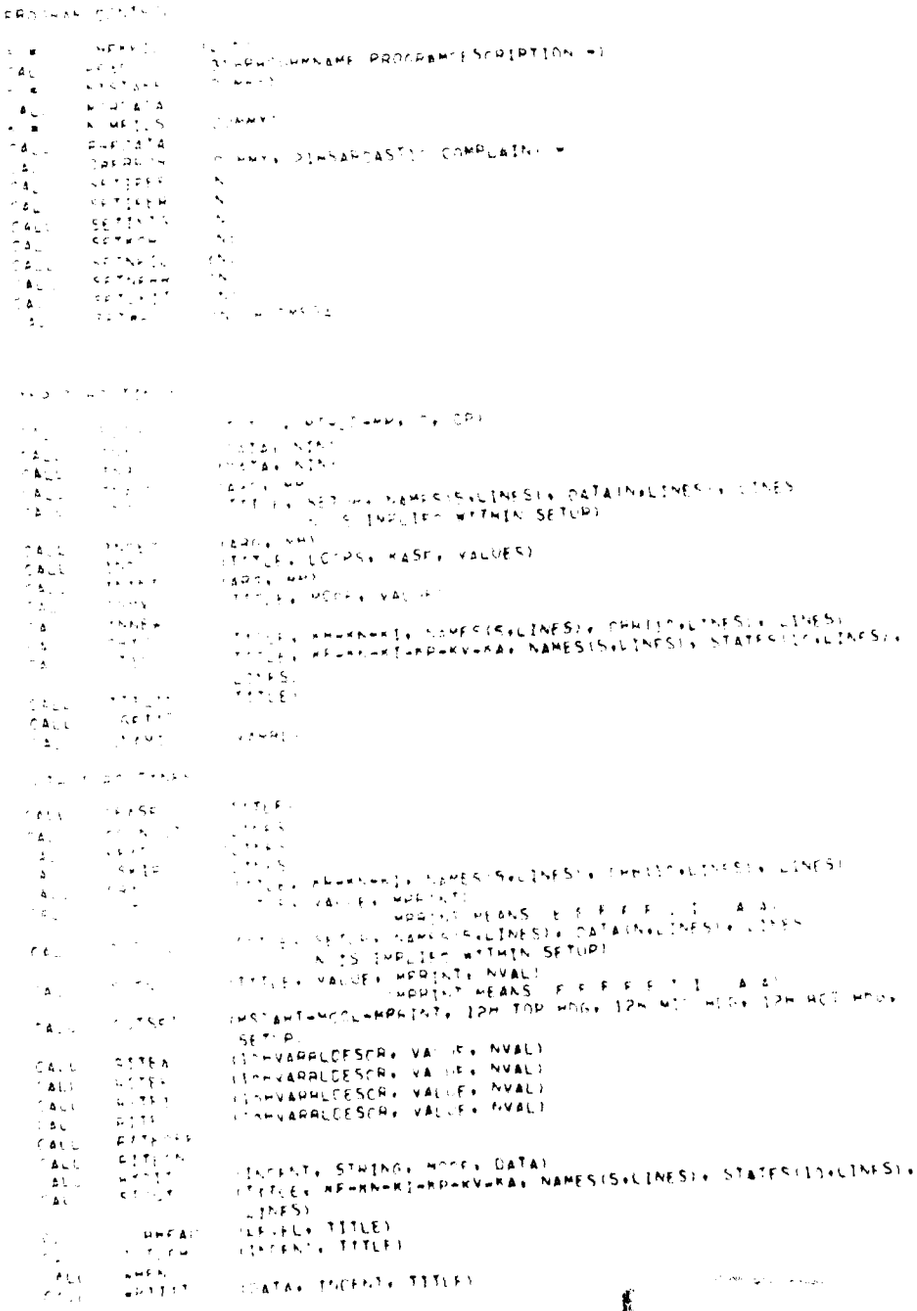
(*,13 or 16+5*N) -- rocket schedule, dimensioned 13 if no maneuvers, 16+5*N if N maneuvers

V(3) -- vector

A, B, AA -- matrices

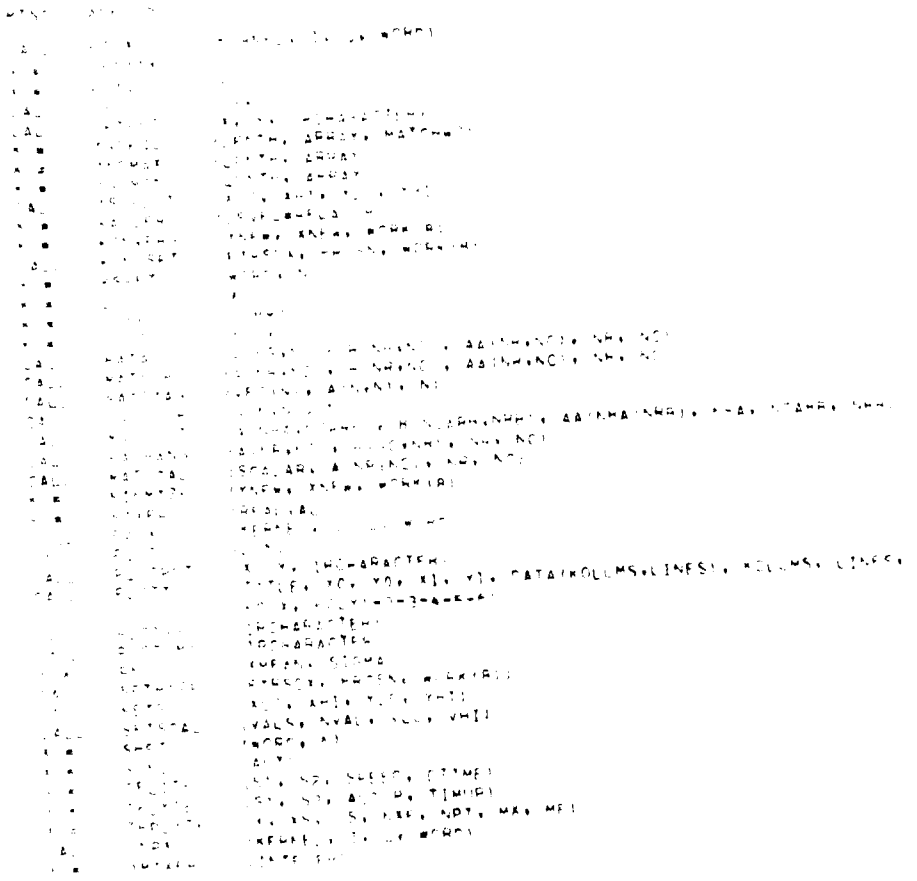
TITLE -- 80 string ending on a dash

SETUP(82) -- array containing format and headings



PRECEDING PAGE BLANK-NOT FILMED

```
VECTOR SUBROUTINES

X =     ALTA            (S)
CALL    ARVEC           (X, V)
X =     ATF             (V)
CAL     CROSS           (V1, V2, V3)
CAL     H, S)           (V1, V2, V3)
X =     DOT             (V1, V2)
X =     ELE             (V)
CALL    EULAN           (NCOEF, AXES(10,3), ANGLES(10), NOERR)
CALL    GRAV            (S, V)
CALL    HPOLAR          (V1, V2, K1, K2, UV(3,3))
CALL    PRO             (V1, V2, V3)
CALL    RSTAR           (TARGET(10,NUM), S, SIGMA(10), TRACK(10,NUM), N,M)
X =     SPEED           (V1, V2)
CALL    STTES           (S, TIME, S2)
CALL    STVEC           (SPECT(10,N), SPOLAR(10,N), N, NOERR(V)
CALL    S HAV           (V1, V2, V3)
CALL    TRVSEM          (S(10,NS), AX(3,3), S2(10,NS), NS, NOERR)
CALL    VEC             (V1, V2)
CALL    VECTR           (X, V1, Y, V2, V3)
CALL    VECTS M         (X, V1, Y, V2, V3)
X =     VMG             (V)


STATISTICS

CALL    ATRAN           (AXES, SLITS, S(10,NSTATE), C, CP, ITS, SCALEN, SCALVAL,
                         SCALOFF, NOUT, NSTATE)
CALL    CRB             (S, V)
CALL    TRDSTB          (S(10,2), TIME, S)
CALL    DSTATS          (S, TIME, S)


UTILITY

CALL    CHEK            (S, CHK)
CALL    ERR             (MODE, VALUE, S1, S2, OMB)
CALL    ERR             (MODE, VALUE, CHK, S)
CALL    ERRSET          (ERR, TIME, ISTATUS, A
X =     GETERR          (MODE, VALUE, A, E, DP)
X =     GETTIME         (MODE, VALUE, CHK)
X =     GETVAL          (MODE, VALUE, S1, S2, OMB)
```



.4 OPTIONAL USER DEFINED ROUTINES

```
ORBEL    1.   NOT USED
ORBEL    2.   LONGITUDE OF ASCENDING NODE
ORBEL    3.   INCLINATION OF ORBIT PLANE
ORBEL    4.   ARGUMENT OF PERIGEE
ORBEL    5.   SEMI-LATUS RECTUM
ORBEL    6.   ECCENTRICITY
ORBEL    7.   TIME AT PERIGEE
ORBEL    8.   PERIOD/(2*PI)
ORBEL    9.   SEMI-MAJOR AXIS
ORBEL   10.   NOT USED

STATE    1.   TIME
STATE    2.   X
STATE    3.   Y-POSITION
STATE    4.   Z-POSITION
STATE    5.   X-VELOCITY
STATE    6.   Y-VELOCITY
STATE    7.   Z-VELOCITY
STATE    8.   X-ACCELERATION
STATE    9.   Y-ACCELERATION
STATE   10.   Z-ACCELERATION

MODEVALUE     AS USED IN ORRP AND ORRTIME
          1.0 * I    VALUE = TRUE ANOMALY
          1.0 * I+1  VALUE = RADIUS (INCREASING)
              * I+2  VALUE = RADIUS (DECREASING)
          MODE * 4   VALUE = TIME
          MODE * 5   VALUE = ALTITUDE (INCREASING)
          MODE * 6   VALUE = ALTITUDE (DECREASING)

MODEVALUE     AS USED IN ORRP2
          MODE * 1   VALUE = TIME OF FLIGHT
          MODE * 2   VALUE = FLIGHT TIME EXCESS OVER MINIMUM-
                             FARMAT FLIGHT TIME
          MODE * 3   VALUE = SPEED AT -LAUNCH- POINT
          MODE * 4   VALUE = VELOCITY ELEVATION AT -LAUNCH-
```

PRECEDING PAGE BLANK-NOT FILMED

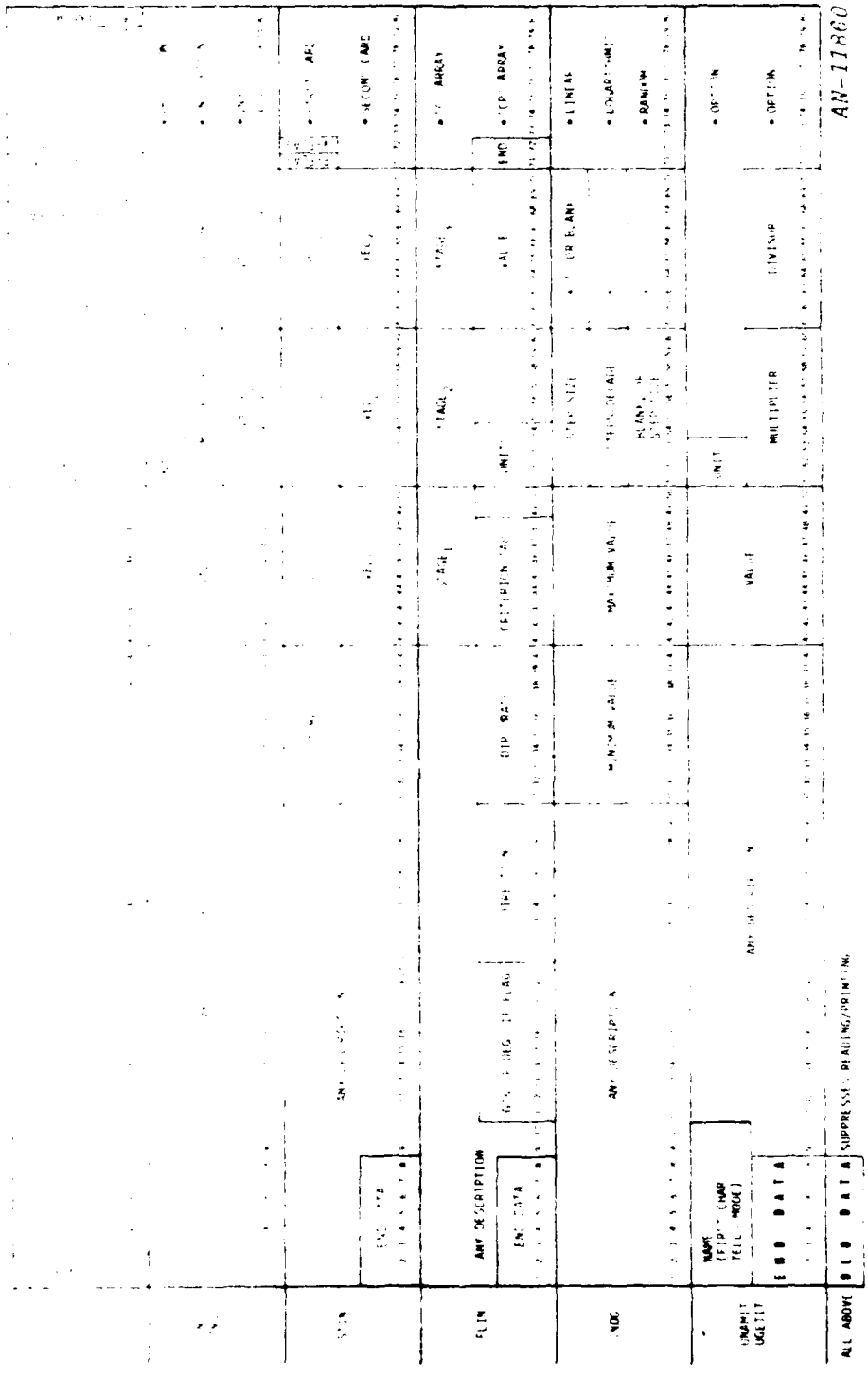
```
C ISTAGE( 3)   TIME OF X-X SEPARATION
C ISTAGE( 4)   X-X SEPARATION TIME
C ISTAGE( 5)   TOTAL MASS AT IGNITION
C ISTAGE( 6)   TOTAL MASS AT BURNOUT
C ISTAGE( 7)   BURNING TIME
C ISTAGE( 8)   REFERENCE AREA FOR DRAG
C ISTAGE( 9)   SUBSONIC DRAG COEFF      MACH LE .5
  ISTAGE(10)   TRANSONIC DRAG COEFF     MACH = 1.
C ISTAGE(11)   SUPERSONIC DRAG COEFF    MACH = 3
C ISTAGE(12)   HYPERSONIC DRAG COEFF    MACH GE 10
C ISTAGE(13)   MAX ALLOWED NORMAL ACCEL
C ISTAGE(14)   MAX ALLOWED ATTACK ANGLE
C ISTAGE(15)     ONE OF THE PARAMETER  CN1
C ISTAGE(16)   NORMAL FORCE PARAMETER  CN2
C ISTAGE(17)   ROLL RATE DAMPING FACTOR  ZETA
C ISTAGE(18)   ROLL RATE DAMPED RESONANT FREQ

CP             CURRENT COMMANDED ACCELERATION  (X-COMP)
CP             CURRENT COMMANDED ACCELERATION  (Y-COMP)
CP             CURRENT COMMANDED ACCELERATION  (Z-COMP)
CP        40   CURRENT STAGE NUMBER
CP        50   TIME WITHIN STAGE
CP        60   IGNITION TIME OF CURRENT STAGE
C              SEPARATION TIME
C         80   STAGE 1-2 IGNITION TIME
CP        90   STAGE 1-2 SEPARATION TIME
CP       100   STAGE 2-3 IGNITION TIME
C              STAGE 2-3 SEPARATION TIME
C              IGNITION TIME
C              CURRENT MANEUVER
CP        A    TYPE OF MANEUVER GOAL
C         B      TIME MANEUVER BEGAN
C         C    MANEUVER MAGNITUDE
C         D      TIME OF IGNITION
CP        E    MANEUVER DIRECTION RATE
CP             MANEUVER
CP               IGNITION
C                  FOR SECOND MANEUVER
C         1A       FOR THIRD MANEUVER  ... ETC
```

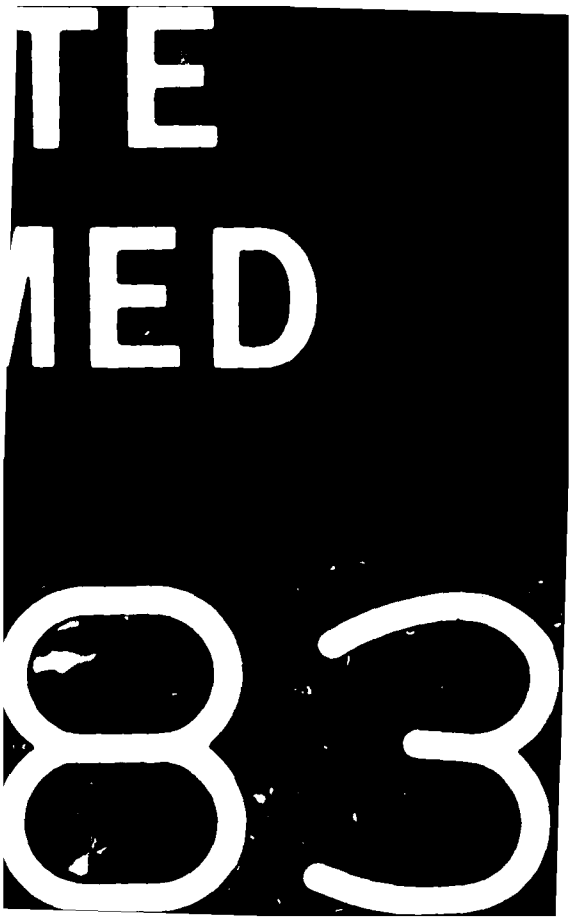
```
[illegible]   ... TAKEN FROM
              FIRST [illegible] CHARS OF SPECIAL CALL OF HEAD
   KPAGE      CURRENT PAGE NUMBER
   LINE       CURRENT LINE NUMBER
   [illegible] CURRENT [illegible]-CHAR TIME OF DAY
 * [illegible] [illegible]-CHAR RUN NUMBER (IN BCD)
 * [illegible] [illegible]-CHAR RUN DESCRIPTION
 * [illegible] [illegible]-CHAR MESSAGE FLAG ... CAUSES MESSAGE
              [illegible] BE READ
 * [illegible] [illegible]-CHAR GENERAL-PURPOSE FLAG ... PLACED
              [illegible]
 * [illegible] [illegible] FROM DATA CARD OR FROM SYSTEM
 * [illegible] [illegible] TIME, FROM DATA CARD OR
              [illegible]
 * [illegible] [illegible], FROM DATA CARD OR 100 PA
 * [illegible] ALLOWABLE LINES PER PAGE, FROM DATA CARD
              [illegible]
              [illegible] OUTPUT ... 0-6 MEANS
              [illegible],SRO,TSRO
 * [illegible] [illegible] RATING GROUP ASSOCIATED WITH
              CLASSIFICATION
 * [illegible] [illegible] DETERMINES UNITS FOR CALCULATIONS ...
              [illegible] METRIC UNITS, #0 MEANS ENGLISH
 * [illegible] FLAG [illegible] INTEGRATION METHOD IN SUBR STINT
              =0 MEANS RECTANGULAR, #0 MEANS RUNGE KUTTA
 * [illegible] FLAG FOR SPECIFYING COORDINATE SYSTEM
              =0 MEANS ORIGIN AT SURF OF FLAT EARTH
              =1 MEANS ORIGIN AT SURF OF ROUND EARTH
              =2 MEANS ORIGIN AT CENTER OF ROUND EARTH
   [illegible] [illegible] WHEN EOF IS FOUND ON 5
              [illegible] CALL EXIT, #0 MEANS RETURN
   [illegible] FLAG INDICATING EOF HAS BEEN FOUND ON 5
 * [illegible] [illegible] ON FIRST DATA CARD, READ BY HEAD

   [illegible] VALUE OF PI (=3.14...)
   [illegible] [illegible] RADIANS TO DEGREES (=180/PI)
              [illegible] DISTANCES ...
              KILOMETERS TO METERS (=1000) IF KUNITS=0
              NAUTICAL MILES TO FEET (=6080) IF KUNITS#0
   [illegible] [illegible] METERS TO FEET (=3.28...)
   [illegible] [illegible] KILOGRAMS TO POUNDS (=2.2...)
   [illegible] RADIUS OF EARTH ...
              [illegible] IN METERS IF KUNITS=0
              [illegible] IN FEET IF KUNITS#0
              ACCELERATION DUE TO GRAVITY ...
              =9.8... IN M/SEC2 IF KUNITS=0
              =32.... IN FT/SEC2 IF KUNITS#0
   [illegible] GRAVITATIONAL CONSTANT = GACC*RBODY**2
   [illegible] ANGULAR SPEED OF EARTH (=7.3E-5) RAD/SEC
   [illegible] ATMOSPHERIC DENSITY AT SEA-LEVEL ...
              =1.225...) IN KG/M3 IF KUNITS=0
              =0.076...) IN LB/FT3 IF KUNITS#0
   [illegible] VALUE OF 2*PI (=6.28...)
   [illegible] VALUE OF .5*PI (=1.57...)
```

5.6 ENTRY-NAME REFERENCE

[illegible] NAME	[illegible] NAME
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	SETPLOT
[illegible]	SETPLOT
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	SETPLOT
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	SETPLOT
[illegible]	SETPLOT
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	MISTAKE
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	MISTAKE
[illegible]	CHEKFIL
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

PRECEDING PAGE BLANK-NOT FILMED




TE
MED
83